Gestion de la sécurité des réseaux à l'aide d'un service innovant de Cloud Based Firewall

Fouad Amine Guenane

Thèse de doctorat de l'Université Pierre et Marie Curie

Ecole doctorale Informatique, Télécommunication et Electronique de Paris

Laboratoire Informatique de Paris 6- Equipe PHARE

Gestion de la sécurité des réseaux à l'aide d'un service innovant de Cloud Based Firewall

Par Fouad Amine Guenane

Thèse de doctorat d'Informatique

Présentée et soutenue publiquement le 13/10/2014

Devant un jury composé de :

A. Bouabdallah, *Professeur à l'Université de Technologie de Compiègne*	Rapporteur
F. Krief *Professeur à L'école nationale supérieure d'électronique, informatique, télécommunications, mathématique et mécanique de Bordeaux*	Rapporteur
D. Tandjaoui, *Professeur au Centre de Recherche sur l'Information Scientifique et Technique (CERIST)*	Examinateur
Marcelo Dias De Amorim, *Directeur de Recherche CNRS*	Examinateur
Pascal Urien, *Professeur à Télécom Paris-Tech*	Examinateur
Mathieu Bouet, *Ingénieur spécialiste réseau, Thales Sécurité et Communication*	Examinateur
Guy Pujolle, *Professeur à l'Université Paris-6*	Directeur de Thèse
Michele Nogueira, *Professeur à l'Université Fédérale du Paraná*	Encadrante

Remerciements :

Je tiens à remercier très sincèrement toutes les personnes qui, par leurs conseils et leurs encouragements ont contribué à l'aboutissement de ma thèse de doctorat :

En premier lieu, je tiens à exprimer ma profonde reconnaissance à mon directeur de thèse, Monsieur le Professeur Guy Pujolle pour m'avoir accueilli au sein de son équipe Phare et du laboratoire Paris 6 et également pour m'avoir fait bénéficier de vos connaissances. Je vous remercie pour votre disponibilité, votre confiance et vos grandes qualités pédagogiques et scientifiques. Votre écoute patiente et vos précieux conseils ont largement contribué au bon déroulement de ce travail.

J'adresse également mes remerciements les plus chaleureux à Madame Michele Nogueira, Professeur à l'Université fédérale du Paraná (Curitiba, brésil) ; vous m'avez guidé et encadré avec un enthousiasme permanent ; vous m'avez transmis les notions nécessaires et indispensables à ma recherche.

Ma gratitude s'adresse également à Monsieur Marcelo Dias de Amorim, Maitre de conférence-HDR à l'Université Paris-6; la richesse et la pertinence de vos remarques, tant sur la forme que sur le fond, ont contribué à améliorer de manière significative le document que je vous soumets aujourd'hui .

Je souhaite aussi remercier les rapporteurs de cette thèse :

- Monsieur Abdelmadjid Bouabdallah, Professeur à l'Université de technologie de Compiègne.
- Mme Francine Krief, Professeur à l'école nationale supérieure d'électronique, informatique, télécommunications, mathématique et mécanique de Bordeaux.

Pour l'intérêt qu'ils ont porté à mon travail.

J'associe à ces remerciements :

- Monsieur Djamel Tandjaoui chercheur au Centre de recherche sur l'information scientifique et technique, CERIST
- Monsieur Pascal Urien Professeur à Télécom Paris Tech.
- Monsieur Abdelkrim Meziane chercheur au CERIST.
- Monsieur Mathieu Bouet spécialiste réseau au sein du groupe Thales sécurité et communication.

Pour avoir accepté d’examiner mon travail.

J’exprime ma profonde reconnaissance envers l’ensemble des professeurs de l’Université de Paris 6 pour la qualité de leur enseignement ; j’ai également beaucoup apprécié la gentillesse de tous les membres du personnel du Laboratoire d’Informatique de Paris 6.

Dédicaces :

A mes chers parents, que Dieu vous garde auprès de moi

A mon Unique frère Anis,

A ma tendre sœur Yasmine,

A mes grand-mères qu'ALLAH leur accorde sa Miséricorde,

A mon grand-père Si EL-FODIL qu'ALLAH lui accorde sa Miséricorde,

A mes fidèles amis,

A ma belle-famille,

A tous mes proches,

Mais surtout à ma femme Lilia qui m'a supporté tout au long de ce travail

Fouad Amine.

Résumé:

Le Cloud Computing a évolué au cours de la dernière décennie, passant d'un simple service de stockage à des services plus complexes, en proposant le software comme service (SaaS), les plateformes comme service(PaaS) et très récemment la sécurité comme service (SECaaS).Dans notre travail, nous sommes partis de l'idée simple d'utiliser les ressources offertes par le Cloud avec un faible coût financier pour proposer des nouvelles architectures de service de sécurité.

La sécurité des environnements virtuels est un sujet majeur pour le déploiement de l'usage du Cloud. Malheureusement, comme ces environnements sont composés d'un ensemble de technologies déjà existantes, utilisées d'une manière nouvelle, de nombreuses solutions sécuritaires ne sont que des solutions traditionnelles reconditionnées à la problématique Cloud et réseaux virtuels.

Le travail effectué dans le cadre de cette thèse vient répondre à la limitation de ressources des équipements physiques de sécurité comme les Firewalls et a pour objectif de proposer de nouveaux services de sécurité composés d'architectures de gestion de la sécurité des réseaux dans le Cloud basé sur le modèle Security as a Service, ainsi que des architectures de management de ces services.

Nous avons pris l'initiative de proposer une architecture totalement Cloud-Based. Cette dernière, permet à un Cloud provider de proposer un service de Firewalling à ses clients. Celui-ci leur demande de s'abonner à l'offre en leur garantissant le traitement (analyse) d'une capacité de bande-passante de trafic avec des règles de filtrages fonctionnelles et d'autres proposées par l'abonné.

Les résultats obtenus ont démontré les aptitudes de nos architectures à gérer et à faire face à des attaques réseaux de type DDoS et à augmenter la capacité d'analyse en distribuant le trafic sur plusieurs pare-feu virtuels.

Summary:

Cloud computing has evolved over the last decade from a simple storage service for more complex services, offering the software as a service (SaaS) platforms as a service (PaaS) and most recently the security as a service (SECaaS). In our work, we started with the simple idea to use the resources offered by the Cloud with a low financial cost to propose new architectures of security service.

The security of virtual environments is a major issue for the deployment of the use of the Cloud. Unfortunately, these environments are composed of a set of already existing technologies used in a new way, many security solutions are only traditional reconditioned solutions to solve the Cloud and virtual networks security issues.

The work done in this thesis is a response to the resource limitations of physical security devices such as firewalls and propose new security architectures consist of management of network security in the cloud-based services following Security as a Service model and propose novel architectures for managing these services.

We took the initiative to propose a completely Cloud-Based architecture. The latter allows a cloud provider to provide firewalling service to its customers. It asks them to subscribe to the offer by guaranteeing treatment (analysis) with a capacity of bandwidth traffic with functional filtering rules and other proposed by the subscriber.

The results demonstrated the ability of our architecture to manage and cope with network DDoS attacks and to increase analytical capacity by distributing traffic over multiple virtual.

Sommaire :

I. Introduction générale:

1. Contexte et motivation:

La puissance apportée par le Cloud et ses machines virtuelles est très peu utilisée aujourd'hui pour gérer la sécurité des entreprises. L'objectif de cette thèse est simple : utiliser cette puissance dans de nouveaux outils et services Cloud très innovants par leur potentiel pour sécuriser les entreprises et en même temps développer les mécanismes nécessaires à la sécurisation des machines virtuelles.

En ce début des années 2010, le Cloud a révolutionné le modèle économique des entreprises. Selon une étude de KPMG, « Innover et tirer profit des ruptures technologiques », le « cloud computing », les applications mobiles et les réseaux sociaux seront les trois innovations les plus structurantes dans les années à venir pour les entreprises. Le Cloud a conforté le concept de serveur virtuel : ce sont des entités logiques qui s'installent sur des machines physiques en général génériques regroupées dans des datacenters. La puissance des datacenters permet de placer plusieurs milliers voire bientôt millions de serveurs virtuels. Ces serveurs virtuels peuvent migrer en fonction de critères qui peuvent être la performance, la consommation électrique, le coût ou d'autres paramètres très variés. Aujourd'hui, les réseaux passent également au stade de la virtualisation: sur une machine physique plusieurs routeurs virtuels peuvent y être installés comme le montre la Figure 1. Ces routeurs virtuels peuvent également migrer en fonction de critères spécifiques aux réseaux : coût, énergie, fiabilité, contrôle de congestion, etc. Cette notion de routeurs virtuels s'étend simplement à tous les équipements réseaux comme les commutateurs, les *Label Switch Routers* MPLS, les box, les passerelles, les serveurs SIP, les PABX, etc. mais doit également pouvoir s'adapter aux équipements de sécurité des réseaux comme les *firewalls,* les IPS-IDS, les serveurs d'identité ou les serveurs d'authentification.

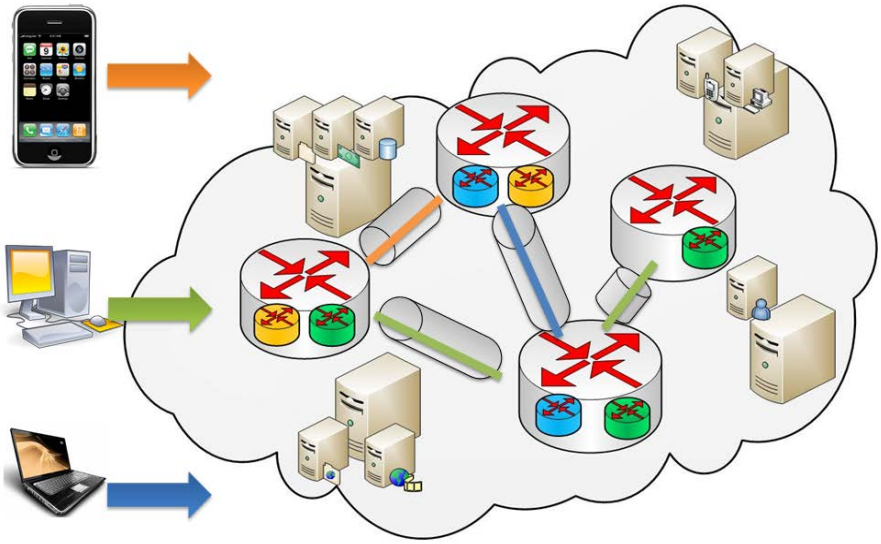

Figure 1- Convergence des réseaux virtuels et des services Cloud

La sécurité des environnements virtuels est un sujet majeur pour le déploiement de l'usage du Cloud. Malheureusement, comme ces environnements sont composés d'un ensemble de technologies déjà existantes, utilisées d'une manière nouvelle, de nombreuses solutions sécuritaires ne sont que des solutions traditionnelles reconditionnées à la problématique Cloud et réseaux virtuels. Cette situation forme un obstacle au déploiement du Cloud.Pour cette raison, notre travail s'intéresse à harmoniser et proposer une nouvelle approche et service sécuritaire compatible avec les infrastructures existantes et la virtualisation, notamment par :

- l'utilisation de machines virtuelles de sécurité : les serveurs d'authentification et les firewalls.
- la conception, réalisation et validation de nouveaux services de sécurité Cloud-Based
- la sécurité des machines virtuelles et plus particulièrement la gestion de leur identité et de leur isolation surtout lors de leur migration dans le réseau (intra et inter-Cloud).

Nous nous intéresserons à l'utilisation de la puissance du Cloud pour gérer la partie calcul des *firewalls*. En effet, dans les caractéristiques des *firewalls* matériels, on précise le débit des interfaces physiques mais on oublie souvent que le débit réel est déterminé par la puissance de calcul disponible dans la machine supportant le *firewall*. Notre travail de recherche propose d'améliorer à la fois la vitesse de calcul et la capacité

de détection des anomalies en augmentant la puissance de calcul des *firewalls* de plusieurs ordres de grandeur. Cette puissance de calcul supplémentaire s'obtient par le concept de *firewalls* virtuels utilisant les immenses ressources qu'offre le *Cloud*.

Dans ce but, nous avons déployé des firewalls virtuels avec deux principales architectures "hybride" et "Cloud-Based". Ces architectures viennent en complément des firewalls physiques traditionnels. Cette thèse propose donc une nouvelle génération de services de Firewalling, dans laquelle les parties virtuelles peuvent se déplacer pour éviter des attaques et optimiser leur fonctionnement dans le Cloud.

2. Contributions:

Le travail effectué dans le cadre de cette thèse a pour objectif de proposer de nouveaux services de sécurité dans le Cloud basé sur le modèle Security as a Service, ainsi que des architectures de management de ces services. Pour cela, trois types d'architecture ont été proposés: Hydride, Cloud-Based et Autonomique.

a. Architecture Hybride de gestion de service de Firewalling Cloud Based:

Nous avons travaillé sur l'amélioration des performances en augmentant la puissance de calcul des pare-feu physiques. L'innovation de notre travail est de proposer une architecture hybride basée sur le modèle « Security As A Service » (SecaaS) fourni par le Cloud. Notre architecture est hybride car elle est constituée de deux principales parties : (i) la partie virtuelle localisée dans le Cloud et (ii) la partie physique. Ce service est considéré comme une ressource supplémentaire qui complète celles déjà existantes. Les résultats obtenus présentent une amélioration significative des performances systèmes et réseaux.

b. Architecture de Management de la sécurité pour un service de Firewalling Cloud Based:

Notre travail est de proposer un service de Firewalling totalement Cloud-Based, qui prend en compte le management de toutes les opérations réseaux et sécurité du service. Cette architecture possède les mêmes caractéristiques que le Cloud, flexibilité et disponibilité. Ce service proposé par un Cloud Provider vient en renfort des pare-feu traditionnels ce qui permettra de faire face au volume important de trafic. Ainsi, cette architecture fournit les ressources nécessaires aux utilisateurs pour faire face à des attaques DDoS mais aussi pour gérer les performances et la fiabilité du service selon leurs besoins. Ceci permet de rapprocher l'expérience du Cloud Provider avec

l'expertise métier de l'utilisateur. Les résultats obtenus ont démontré les aptitudes de ce service à faire face à des attaques réseaux de type Flooding et à augmenter la capacité d'analyse en distribuant le trafic sur plusieurs pare-feu virtuels.

c. Système multi-agents pour le management des opérations du service Firewalling Cloud-Based:

Nous présentons un système multi-agents pour la gestion et la répartition de trafic dans le service Firewalling. Cette architecture virtuelle prend en charge la création et la suppression automatisée des firewalls virtuels en fonction du trafic à analyser, fournissant ainsi une gestion dynamique et à la demande de l'architecture de sécurité présentée précédemment. Notre approche est basée sur un système multi-agents qui collecte et agrège toutes les informations distribuées tant au niveau de la passerelle et des firewalls virtuels. Le choix d'un modèle multi-agents améliore la réactivité et l'adaptabilité rapide aux fluctuations et aux changements dans l'environnement Cloud. Notre système est adaptatif et laisse la possibilité de mettre en place des optimisations futures, que cela soit au niveau des agents du protocole de communication ou bien des algorithmes utilisés.

3. Plan de Thèse :

La thèse est structurée de la manière suivante :

Le deuxième chapitre présente un état de l'art sur la virtualisation des équipements et services de sécurité dans le Cloud. La première partie introduit les services de sécurité qui sont déployés à nos jours dans le Cloud pour diverses utilisations réseaux, mobiles ou applicatives. Ainsi, la deuxième partie s'intéresse à la technologie firewall : Concept, architecture, fonctionnalité ainsi que leurs limitations, pour ensuite arriver à leur déploiement dans le Cloud. Enfin, nous discuterons l'utilisation des systèmes multi-agents dans la gestion de tels services de sécurité dans le Cloud.

Le troisième chapitre décrit notre première contribution. Nous présenterons tout d'abord notre architecture hybride avec tous ses composants suivie de deux modèles de déploiements le « Secure Forwarding » et le « Secure Sharing ». Nous proposerons ensuite de tester notre architecture avec ses mécanismes de déploiement sur des cas d'utilisations ce qui nous permettra de montrer son efficacité et son impact sur les performances.

Le quatrième chapitre de cette thèse est dédié à proposer une architecture de management de service Firewalling exclusivement basé dans le Cloud. Puis, une présentation de l'architecture générale, de son déploiement et des opérations réseaux qui la caractérisent. Une partie des cas d'utilisations et discussions des tests détaille et discute les résultats obtenus lors des scénarios des tests pour démontrer la robustesse et les performances élevées de la solution.

Le cinquième chapitre propose une architecture de management de service basée sur un système multi-agents. Ce dernier est présenté avec son architecture générale en définissant les agents utilisés, leurs fonctionnalités ainsi que leur protocole de communication. L'architecture logicielle reprend les grandes lignes de l'architecture générale puis détaille l'implémentation de ce système via la plateforme Jade.

Enfin, le sixième chapitre présente une conclusion générale dans laquelle nous effectuons un bilan des contributions réalisées et nous identifions les perspectives de notre travail.

II. Etat de l’art sur la virtualisation des équipements et services de sécurité réseaux:

On définit la virtualisation comme un ensemble de techniques visant à faire fonctionner plusieurs systèmes d'exploitation sur le même support physique en partageant les ressources de ce dernier. Aujourd'hui, cette approche remporte un immense succès car elle permet d'améliorer la fiabilité, la personnalisation, la flexibilité des environnements et l’indépendance de l’emplacement et des accès à la demande. Dans des domaines comme les Datacenter et les réseaux, la virtualisation est vue comme la solution aux problèmes que sont l’optimisation des performances, l’économie d’énergie ou le coût de déploiement de nouveaux équipements, et grâce aux principes de mutualisation et de partage des ressources physiques entre différents clients.

La puissance apportée par le Cloud et ses machines virtuelles est très peu utilisée aujourd’hui pour gérer la sécurité des entreprises. L’objectif de ce travail est d’utiliser cette puissance dans de nouveaux outils et services Cloud très innovants par leur potentiel pour sécuriser les entreprises.

Aujourd'hui, le Cloud Computing devient de plus en plus populaire comme un modèle Pay-as-You-Go pour fournir des services à la demande sur Internet. Il permet aux entreprises de mieux gérer leurs ressources en profitant d’un service à la demande moins cher, ce qui permet de réduire le coût de l'investissement dans de nouvelles ressources locales. La question qui se pose alors est : « Comment exploiter les avantages de la flexibilité de la gestion des ressources qu’offre le Cloud pour augmenter la puissance de calcul des firewalls physiques pour faire face à la montée en charge du trafic de l’entreprise ? ». En effet, ceci impacte de manière significative les performances du firewall physique, le transformant en goulot d’étranglement pour le réseau sans avoir à investir dans l’achat de matériel.

Dans cette partie, nous allons discuter les solutions de sécurité Cloud-Based pour avoir une vue d’ensemble de ce qui est proposé. Puis, nous préciserons notre démarche en nous intéressant aux déploiements des firewalls principalement dans les environnements virtualisés, et ce en discutant les avantages existants par rapport aux pare-feu physiques et leurs déploiements, mais aussi leurs limites.

1. Etat de l'art des solutions de sécurité « Cloud-Based » :

Alors que le Cloud Computing augmente l'agilité, l'évolutivité et l'efficacité de l'entreprise, il introduit cependant de nouveaux problèmes de sécurité. En effet, les solutions traditionnelles de sécurité deviennent obsolètes car la majorité du trafic des réseaux virtuels ne quitte pas nécessairement le serveur physique [1] et ne permet pas donc un contrôle permanent , en plus du fait de l'augmentation du trafic crypté qui impacte les pare-feu physiques et les limite dans leurs fonctionnalités [2].

Dans le but de pallier à ces limitations, un nouveau type de services de Cloud Computing émerge et de nombreux éditeurs de sécurité tirent systématiquement parti des modèles de Cloud Computing pour offrir des solutions de sécurité (exemple d'analyse anti-virus en ligne, de firewall virtuel…etc.).

La thématique de recherche engendrée par l'utilisation des caractéristiques du Cloud pour proposer de nouveaux mécanismes afin de prévenir et/ou de protéger les entreprises des différents types d'attaques est porteuse et totalement nouvelle. La nécessité de mettre en place des services de sécurité basés sur le cloud est discutée par Modi et al. [3].

Yu et al. [4] ont démontré que la puissance du Cloud Computing peut être utilisée pour contrer des attaques DDoS (Distributed Denial of Service) en se servant du principal avantage du Cloud par rapport à ce type d'attaques c'est-à-dire l'extensibilité des ressources. De là, ils ont mis en place une solution d'atténuation d'attaque DDoS basée sur un mécanisme d'allocation dynamique de ressources. Il s'appuie sur l'allocation de ressources supplémentaires mises à disposition spécifiquement pour alimenter la solution.

Yassinet al. [5] a proposé un Framework de service de détection d'intrusion qui repose sur le modèle Software as a Service (SaaS) qui est capable de localiser des activités malveillantes dans le réseau, et d'améliorer les limitations des solutions de détection d'intrusion classique. Houmansadr et al. proposent dans [6] un outil de détection d'intrusion totalement Cloud-Based pour smartphone, qui effectue en mode connecté en permanence une analyse en profondeur des comportements des applications installées sur le smartphone pour détecter toute mauvaise conduite. Cette proposition assure la sécurité de l'appareil malgré des ressources très limitées. Dans l'article [7], Menget al, conçoivent un modèle parallèle d'adaptation de la signature pour la détection

d'intrusion dans un environnement Cloud ; leurs expériences ont prouvé que leur modèle proposé peut atteindre des performances prometteuses quand il est totalement Cloud-Based. Selon le même auteur (Meng) dans l'article [8], on arrive à réduire les faux positifs ainsi que la surcharge du système si nous utilisons le modèle « Cloud as a Service », car il peut fournir une puissance de calcul suffisante pour cette tâche.

Darwish précise dans son article [9], qu'en raison de la nature du cloud computing, les méthodes pour prévenir ou arrêter les attaques DDoS sont très différentes par rapport à celles utilisées dans les réseaux traditionnels. Dans ce contexte, [10] propose un testbed Cloud-Based qui permet aux opérateurs d'émuler différentes topologies de réseau, des services, et d'analyser les attaques qui menacent ces systèmes, ce qui facilite l'examen du comportement du réseau sous attaque.

Il est clair que fournir des solutions de sécurité adaptées pour des systèmes distribués complexes est d'une importance primordiale. Le modèle Security-as-a-Service est le concept qui doit être mis en œuvre dans ce type d'architecture. Getovetal. [11] donne un aperçu des principaux enjeux et défis de la sécurité dans le Cloud. Il discute les solutions existantes et proposées avec une attention particulière à la sécurité comme une approche de service. Certaines directions disponibles pour les travaux futurs sont également abordées.

Plusieurs utilisations du modèle SECaaS ont également été publiées ; l'une d'elles a été présentée par Krishnan et al.[12], qui propose un service de gestion de la sécurité (SMAS) à travers laquelle les utilisateurs et les fournisseurs de cloud peuvent profiter des solutions de gestion des identités et d'accès, de confidentialité, d'audit et de comptabilité. La caractéristique de ce travail est que les services peuvent être utilisés à la demande

La Security-as-a-Service pourrait être présentée sous la forme d'une architecture Cloud-Based pour mener des actions de sécurisation dans le domaine de la cyber-sécurité tel que présenté dans le document de Yuet al. [4] qui tire parti de l'infrastructure Cloud.

Dans [13], Amoroso affirme que les pare-feu virtuels de type Cloud-Based sont considérés comme la prochaine grande tendance des fonctionnalités de pare-feu pour la protection des infrastructures nationales d'informations.

La plupart des travaux mentionnés ci-dessus sont très intéressants pour assurer la sécurité de l'entreprise au moyen de solutions Cloud-Based.

La proposition de J. Esler and al.[14], de mutualiser plusieurs services de sécurités (firewalling, IPS-IDS, filtrage web, anti-spam…etc.) faite en 2009 dans une même unité matériel, grâce à des techniques de virtualisations, représente une solution de départ pour faire face aux limites des systèmes traditionnels de détection d'intrusions.

Cependant, la recherche sur les attaques DDoS et les moyens de s'en prémunir dans un environnement Cloud est encore à un stade précoce. La disponibilité d'un service de sécurité dans le Cloud couvre divers aspects, tels que :

- les stratégies d'atténuation des attaques DDoS [15] ou EDoS (Economic Denial of Sustainability) [16] dans un environnement Cloud.
- DDoS défense comme un service dans le Cloud [17]
- Des architectures de sécurité contre les attaques DDoS dans le Cloud [18].

Néanmoins, la gestion de pare-feu Cloud-Based est encore un gros problème. Les questions qui se posent sont:

- Comment instancier, supprimer et migrer les pare-feu virtuels dans le Cloud?
- Comment gérer le trafic entre les pare-feu physiques situés dans l'entreprise et d'autres virtuels dans le Cloud?
- Comment gérer la distribution des règles de pare-feu afin d'éliminer tout conflit de règles de sécurité?
- Comment allouer des ressources adéquates et suffisantes pour chaque pare-feu virtuel dans le Cloud?

Ce document portera sur ces questions. L'accent est principalement mis sur la gestion des flux entre les différentes entités, ainsi que l'élasticité des ressources pour gérer l'approvisionnement et la réallocation des ressources de manière autonome.

2. Background et Etat de l'art sur les solutions de Firewalling :

Un Pare-feu est la première et principale ligne de défense pour les services et applications d'un réseau. Les performances d'un pare-feu dépendent fortement de ses capacités matérielles et de la disponibilité de ses ressources. Si par exemple un pare-feu est submergé lors d'une attaque DDoS, l'accès aux applications de l'entreprise n'est plus possible. Donc, pour faire face à cette situation cruciale, les entreprises sont

obligées d'investir massivement dans l'achat d'équipements supplémentaires. Cette situation impose des coûts importants pour la plupart des entreprises, notamment les petites et moyennes entreprises. Selon [19] et [20], le coût de déploiement et d'entretien d'un pare-feu physique est estimé à 116.075$ pour la première année et un coût annuel de 108.200$ pour une société américaine de taille moyenne caractérisé par 5Mbps de connectivité Internet. De plus, des coûts supplémentaires sont engagés pour l'embauche, l'entretien, la surveillance et les mises à jour.

Les politiques de sécurité du pare-feu sont des règles de filtrage ordonnées qui définissent les actions effectuées sur les paquets afin de satisfaire à des conditions spéciales. Il existe trois principaux types de pare-feu[21][22][23]:

- Filtrage de paquets : plus communément appelé « Packet-Filter », les routeurs possèdent cette fonctionnalité qui consiste à examiner les paquets au niveau des couches réseau et/ou transport, permettant à des paquets seulement autorisés à être transférés. Il possède quelques avantages : faible surcharge de trafic à haut débit et peu coûteux. Cependant, son niveau de sécurité est très faible.
- Stateful Inspection : fournit une capacité de filtrage au niveau applicatif tout en fonctionnant au niveau de la couche de transport [24]. Il améliore les fonctions du filtrage de paquets par le suivi de l'état des connexions et le blocage des paquets qui s'écartent de certains Etats. Cependant, cela implique l'utilisation de plus de ressources et plus de complexité pour la gestion des opérations du pare-feu[24].
- Pare-feu applicatif : ce dispositif contient un agent mandataire agissant comme un lien transparent entre deux hôtes qui souhaitent communiquer entre eux. Ce dispositif ne permet pas de connexion directe entre les deux hôtes. Points négatifs : les pare-feu applicatifs ne protègent pas contre les attaques au niveau des couches inférieures ; ils nécessitent un programme distinct pour chaque application et une consommation de ressources très élevée pour des performances limitées.
- Next Generation Firewall (NGFW) : cette technologie représente une évolution par rapport au pare-feu traditionnel [25] en intégrant une variété de fonctions de sécurité comme le filtrage anti-spam, anti-virus, un système de détection ou de prévention d'intrusion (IDS / IPS) dans une plate-forme

intégrée. Les NGFWs fournissent également une inspection plus granulaire et une plus grande visibilité de trafic que celle des pare-feu traditionnels [14]. Les NGWFs font face à une augmentation du trafic ‘crypté’ ce qui met en lumière leurs limitations [26].

Bien que l'utilisation d'un pare-feu présente plusieurs avantages, la capacité de filtrage de l'équipement est le principal inconvénient. Effectivement, cela dépend de son intégration dans le réseau, la complexité de la politique de sécurité et le débit de paquets qui provoquent dans la majorité des cas un déni de service. Certains articles comme celui de Liu [27] proposent des améliorations aux algorithmes et méthodes d'analyse des pare-feu. Cependant, l'évolutivité et le bon fonctionnement de ces algorithmes conduisent à une demande croissante de ressources de pare-feu.

A cet effet, plusieurs études ont validé certains modèles de déploiement qui améliorent et optimisent la capacité de filtrage des pare-feu physiques en modifiant différentes caractéristiques comme l'architecture et la distribution des règles. Il est à noter que la capacité de filtrage d'un dispositif dépend de son intégration dans le réseau. Dès lors, une mauvaise configuration entraîne une perte significative des performances. Deux principales architectures sont proposées :

- Architecture avec pare-feu simple: c’est un modèle qui propose un dispositif de pare-feu matériels qui agit comme un routeur ; sa configuration est une combinaison de filtrage statique des paquets et de blocage basés sur l'autorisation des demandes seulement. En général les performances de ce type de déploiement dépendent de la gamme de pare-feu qui le compose. Par conséquent, le pare-feu porte atteinte aux performances de tout le réseau ralentissant ainsi la fonction de routage qui est sa tâche principale.

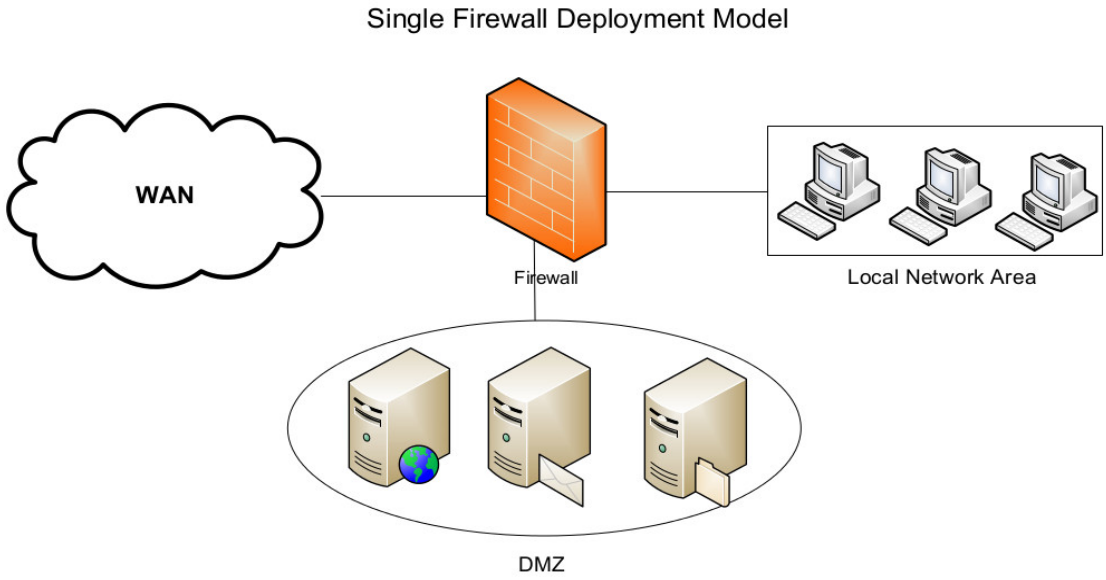

Figure 2 - Architecture de pare-feu simple

- Architecture de pare-feu Distribué : ce modèle est divisé en deux principaux types comme le montre la Figure 3 :
    - Le côté gauche de la figure présente la distribution en cascade [28], l'idée étant de combiner un grand nombre de pare-feu pour augmenter la niveau de sécurité en utilisant différentes heuristiques pour le positionnement pare-feu.
    - La deuxième partie de la figure correspond à la distribution parallèle, appelée souvent pare-feu à équilibrage de charge. C'est une approche qui a été conçue pour augmenter la vitesse d'inspection et de circulation du flux dans le réseau. Le parallélisme est une technique qui améliore la performance du pare-feu. Cependant, les deux modèles existants ne possèdent pas la même distribution d'axe. En effet, l'axe peut correspondre aux paquets (données) ou aux règles de filtrage. De nombreuses études comme [29] et [30]discutent ces deux modèles. Le principal inconvénient qui en découle est le nombre de pare-feu utilisés qui dépasse celui de nœuds à protéger. Dans certains cas, la gestion de tous les pare-feu ajoute également un coût supplémentaire en plus de celui du matériel qui n'est pas négligeable.

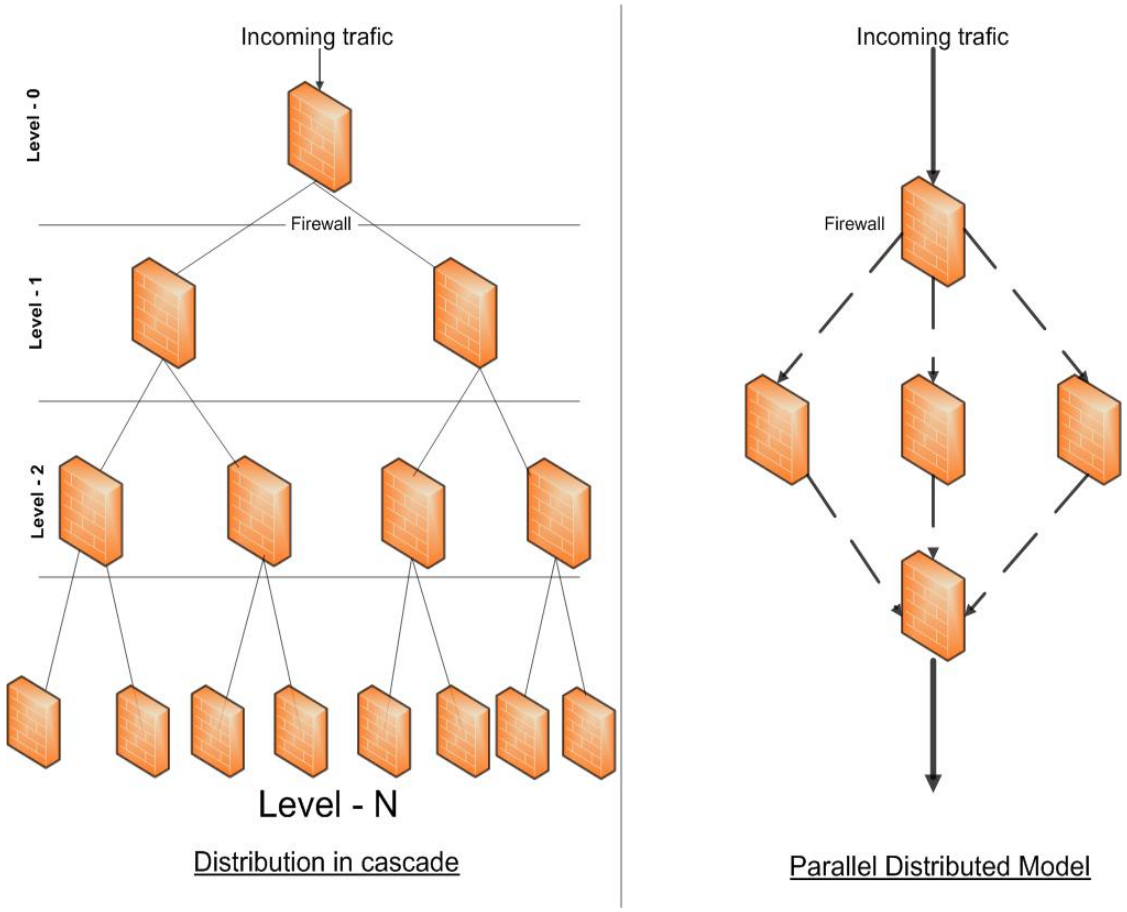

Figure 3- Architecture de Firewalls distribués

Afin de réduire les coûts de gestion et de déploiement des pare-feu, les entreprises externalisent leurs managements pour les fournisseurs de Cloud, dans un modèle de type Software-as-a Service (SaaS) [31][32]. La demande actuelle pour des services plus performants oblige les organisations à déployer et à maintenir des solutions de plus en plus innovantes. Avec l'avènement de la virtualisation et du Cloud Computing, les pare-feu physiques ne sont plus conçus pour inspecter et filtrer la grande quantité de trafic des machines virtuelles[1].

Dans l'industrie, plusieurs développements de solutions de pare-feu virtuels sont proposés. Par définition, C'est des machines virtuelles spécialisées qui fournissent les techniques habituelles de filtrage de paquets et de monitoring[33]. disponibles en deux modes [34]:

- Mode Hyperviseur : c'est un pare-feu virtuel exécuté au niveau de l'hyperviseur et plus précisément dans le 'Virtual Machine Monitor' (VMM). Par conséquent, il n'est pas considéré comme une partie du réseau et il ne gère que le trafic local (intra hyperviseur).

- Mode Bridge : il est positionné comme un bridge entre les différents segments des réseaux virtuels. Par conséquent, c'est une machine virtuelle à part entière. Les performances de ce type de firewall virtuel dépendent bien évidemment des ressources allouées. La migration des machines virtuelles devient problématique pour ce type de pare-feu virtuel, car il doit gérer les différentes politiques de sécurité.

Jusqu'à présent, les fournisseurs de solutions de virtualisation offrent des pare-feu virtuels comme meilleure solution pour l'isolation du trafic et d'analyse de réseau déclinés sous différents noms. Pour Cisco, c'est le Virtual Security Gateway(VSG) réparti sur les nœuds physiques du Cloud, il est considéré comme un pare-feu pour hyperviseur spécifique. VMware a beaucoup investi en termes de sécurité et propose la série de produit 'v-Shield'.

3. Etat de l'art des systèmes multi-agents dans la gestion des ressources des services Cloud-Based :

a. Gestion des ressources dans le Cloud :

L'allocation des ressources a été largement étudiée par rapport aux systèmes d'exploitation, réseau informatique, les fermes de serveurs et la gestion du Cloud Computing. Le but des mécanismes d'allocation est de garantir une application afin de satisfaire les besoins des clients avec l'infrastructure du fournisseur. À cet égard, de nombreuses stratégies d'allocation de ressources ont été proposées, qui mettent l'accent sur différents aspects de la performance. Quelques considérations sont prises en compte lors de l'élaboration d'une stratégie d'allocation de ressources:

- la charge de travail courant du Cloud,
- La minimisation du temps de réponse.
- La maximisation de la satisfaction des demandes
- La minimisation du gaspillage des ressources.

Par conséquent, nous pouvons dire que l'allocation des ressources est un problème d'optimisation[35][36].

Dans l'article [37], les auteurs ont formulé le problème de planification des ressources intra-Cloud en se basant sur un modèle de file d'attente du système. Pour les deux scénarios de service: classe unique et multi-classe, le schéma proposé vise à

minimiser le temps de réponse et le coût des ressources. Différentes stratégies d'allocation des ressources du réseau et leurs applications dans un environnement Cloud Computing ont été présentées dans [38]. Ainsi, ils ont traité les défis majeurs à la progression de l'allocation des ressources dans le Cloud Computing en fonction de plusieurs types de mécanismes d'allocation de ressources.

Une stratégie d'allocation des ressources basée sur le service level agreement (SLA) a été proposée dans [39] et qui vise à améliorer la consommation des ressources des centres de données volumineux tout en fournissant des services avec un bon niveau de QoS pour les consommateurs.

Un mécanisme de ressources de l'allocation pour les machines sur le Cloud a été proposé dans l’article [40]. Ce mécanisme est basé sur les principes de la formation de coalition et le principe d'incertitude de la théorie des jeux. Les auteurs ont comparé les résultats de l'application de ce mécanisme avec des méthodes d'allocation de ressources existantes qui ont été déployées dans le Cloud. Ils ont montré également que cette méthode d'allocation des ressources avec la coalition-formation des machines sur le Cloud ne conduit pas seulement à une meilleure utilisation des ressources, mais aussi à la satisfaction de la demande ultérieure. L'allocation des ressources pour la théorie des jeux dans le Cloud a été discutée dans [41] en suivant deux étapes. Dans la première étape, chaque participant résout son problème d'optimisation indépendamment, sans tenir compte du multiplexage des affectations de ressources. Dans la deuxième étape, un mécanisme évolutif changeant les stratégies multiplexées des solutions optimales initiales (en minimisant les pertes d'efficacité) est conçu.

Un modèle "Utilisation Maximisation" proposé par [42] se concentre sur le problème d'allocation des ressources dans le Cloud. Ils ont utilisé la plateforme « Cloudsim » pour simuler les différentes entités impliquées dans l'environnement de l'allocation des ressources, les interactions et les procédures entre les entités concernées. Ainsi, des algorithmes pour les clients et les courtiers de ressources ont été proposés.

Tous ces algorithmes prennent en compte des paramètres bien spécifiques qui dépendent du besoin ou du service proposé. Ils n’ont pas la possibilité d’adapter leur réaction à des changements d’état, et si c’est le cas cela prend un temps de convergence assez élevé ce qui est impactant. Nous avons choisi de discuter l’utilisation des systèmes

multi-agents dans l'allocation des ressources et la gestion de services Cloud, ceci pour nous permettre de concevoir une solution de management plus autonome et adaptative.

b. Systèmes multi-agents pour la gestion des services Cloud :

Plusieurs études ont établi un modèle mathématique pour modéliser les besoins de provisionnement des ressources dans le Cloud Computing, ces études sont basées sur la théorie des files d'attentes majoritairement. Les chercheurs ont proposé de nombreuses approches multi-agents dans un contexte Cloud en vue de surmonter les limites de ce dernier telles que l'allocation des ressources et les menaces de sécurité. Par exemple, dans l'article [43], les auteurs ont proposé une combinaison de la technologie Agent avec le Cloud pour arriver à une méthode pour le calcul du modèle de gestion des ressources. Il conforte la théorie que l'utilisation d'agents permet d'atteindre efficacement la gestion des ressources dans le Cloud.

Dans l'article [44], Talia montre que l'intégration des SMA dans un contexte Cloud peut permettre de hautes performances pour des systèmes complexes et des applications intelligentes prouvant ainsi qu'on peut obtenir une infrastructure fiable et évolutive sur laquelle on exécute une application à grande échelle. Les SMA ont aussi été utilisés dans le Cloud pour proposer un service de gestion d'accès. En effet, Habiba, M et al, dans leur article [45] soulignent que les systèmes traditionnels ne sont pas suffisamment efficaces pour prendre en charge la fonctionnalité du contrôle d'accès dans le Cloud principalement en raison de la grande extensibilité de l'environnement de cloud. Ils ont utilisé un système multi-agents pour définir l'accessibilité et la fonctionnalité de leur modèle afin d'améliorer le système de contrôle d'accès. D'autres utilisations des systèmes multi-agents pour fournir des services de sécurité Cloud-based ont été proposés comme pour l'article [46] où les auteurs ont utilisé une architecture SMA pour assurer la confidentialité et la disponibilité pour un service de stockage collaboratif hébergé dans le Cloud, et aussi dans un service de gestion des catastrophes naturelles présentées dans l'article [47], Les auteurs ont utilisé un modèle de workflow pour aider et maintenir le sauvetage et la réorganisation des activités en cas de catastrophe.

D'autres travaux ont traité le concept de description formelle des SMA. Leurs études visent à évaluer les fonctionnalités intégrées et à présenter les spécifications formelles de systèmes multi-agents. Typiquement dans l'article [48], Khezami et al proposent un nouveau formalisme de collaboration dans MAS entre les agents combinés

avec le modèle de Ferber, caractérisés par la possibilité de l'auto-évaluation dans l'application de travail collaboratif. Leurs résultats montrent que le formalisme proposé gère plus efficacement la communication de l'agent pour une meilleure production.

Les modèles SMA ont été utilisés pour simplifier et améliorer la gestion des architectures des services Cloud. A notre connaissance, aucun service de Firewalling Cloud-Based n’a été traité. La majorité des recherches se sont concentrées sur les méthodes de synchronisation des politiques et règles de filtrage sur un système de firewall distribué. Notre travail traite de la répartition du trafic dans un environnement Cloud sur plusieurs instances de pare-feu virtuels sans prendre en considération la distribution des règles de filtrage.

4. Conclusion

En conclusion dans cette partie, nous avons présenté les différentes solutions existantes pour le déploiement de pare-feu que cela soit dans des environnements physiques ou virtuels. Les pare-feu physiques sont limités à cause principalement de la limite de leurs ressources physiques, ajouté à cela un coût financier considérable. Les Pare-feu virtuels sont prometteurs, Ils ne sont pas limités par les ressources et permettent un déploiement très dynamique. Cependant, les pare-feu virtuels sont non adaptés et impuissants contre les attaques massives de l'extérieur du domaine virtualisé ce qui compromet leur fiabilité.

III. Architecture Hybride de Management des Opérations Réseau pour les Services de Firewalling Cloud Based :

Nous travaillons sur l'amélioration des performances en augmentant la puissance de calcul des pare-feu physiques. L'innovation de notre travail est de proposer une architecture hybride basée sur le modèle « Security-As-A-Service » (SecaaS) fourni par le Cloud. Notre architecture est **hybride** car elle est constituée de deux principales parties : (i) la partie virtuelle localisée dans le Cloud; (ii) la partie physique représentée par le firewall physique de l'entreprise comme le montre la Figure 4.

La partie virtuelle est composée de machines virtuelles, dans lesquelles sont

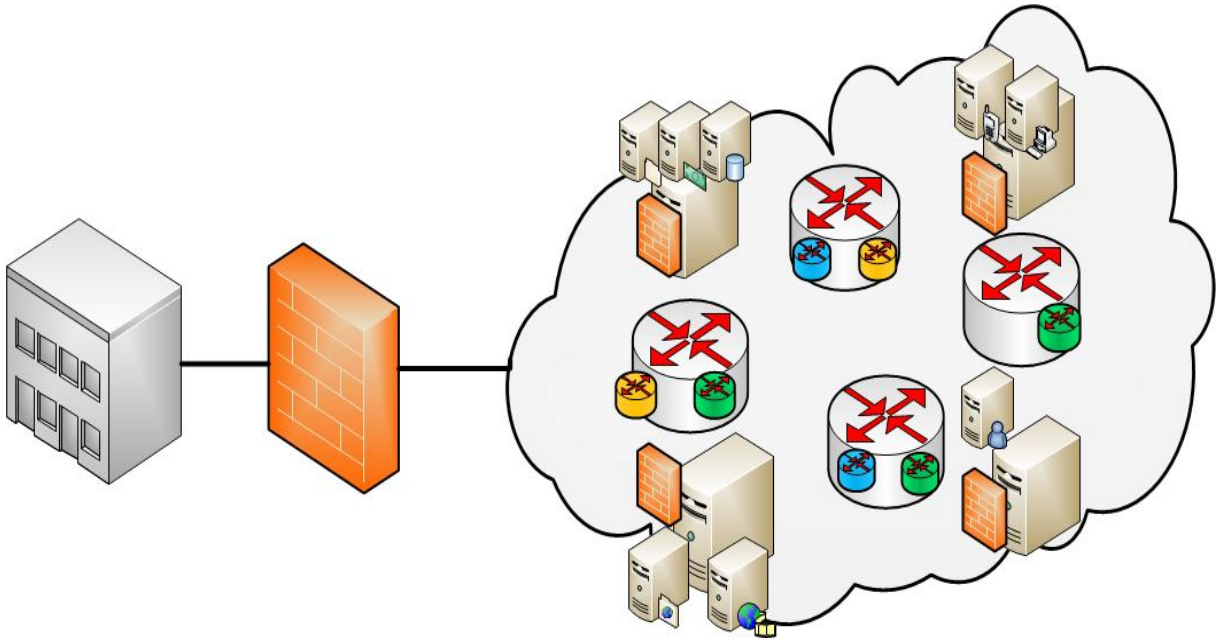

Figure 4- Architecture générale du service firewalling

exécutés des programmes de pare-feu avec de nombreuses fonctionnalités telles que l'analyse, la surveillance, la journalisation de trafic et tout ceci avec un provisionning de ressources dynamiques. La partie physique représente principalement le pare-feu physique.

Lorsqu'un client utilise un service de sécurité offert par le fournisseur de Cloud, ce dernier est considéré comme une ressource supplémentaire qui complète celles déjà existantes ; dans notre cas la ressource physique limitée qui doit être amplifiée pour pouvoir considérer l'augmentation en charge du trafic du client. Le but principal étant de rediriger le trafic destiné au pare-feu physique lorsque celui-ci est surchargé vers le service de firewalling composé de pare-feu virtuels situés dans le Cloud.

Les principales aspirations de notre architecture hybride sont:

- La disponibilité et les performances: en effet, l'utilisation des ressources externes, comme celles fournies par le Cloud permettent d'accroître la disponibilité des services. Nous proposons de partager le trafic destiné au pare-feu physique à d'autres pare-feu virtuels quand une congestion du réseau ou une surcharge des ressources physiques surviennent.
- La prise de décision et la gestion du processus d'équilibrage de charges entre les pare-feu physiques et les pare-feu virtuels sont très importantes. A cette fin, nous avons opté pour un modèle composé de deux parties distinctes : « Physical Firewall Management Center » localisé au niveau du pare-feu physique et « Virtual Firewall Management Unit » déployé dans chaque firewall virtuel, ceci afin d'assurer une adaptation et réaction rapide aux changements de flux ainsi qu'une optimisation de la prise de décision.

Dans ce chapitre, nous allons décrire les composants de notre architecture hybride, puis, nous présenterons deux schémas de déploiement qui nous aideront à valider notre proposition. Nous avons soumis les deux déploiements précédents à deux cas d'utilisation: le cas d'une congestion de réseau et celui d'une attaque de déni de service distribué.

1. Architecture Hybride du Cloud-Based Firewalling Services:

Nous employons une approche fondée sur deux principales caractéristiques du Cloud Computing qui sont : le dynamisme et l'instanciation à la demande dans le but d'améliorer les performances. Nous atteignons cet objectif en augmentant la puissance de calcul de pare-feu physiques et en les adaptant à l'augmentation du trafic.

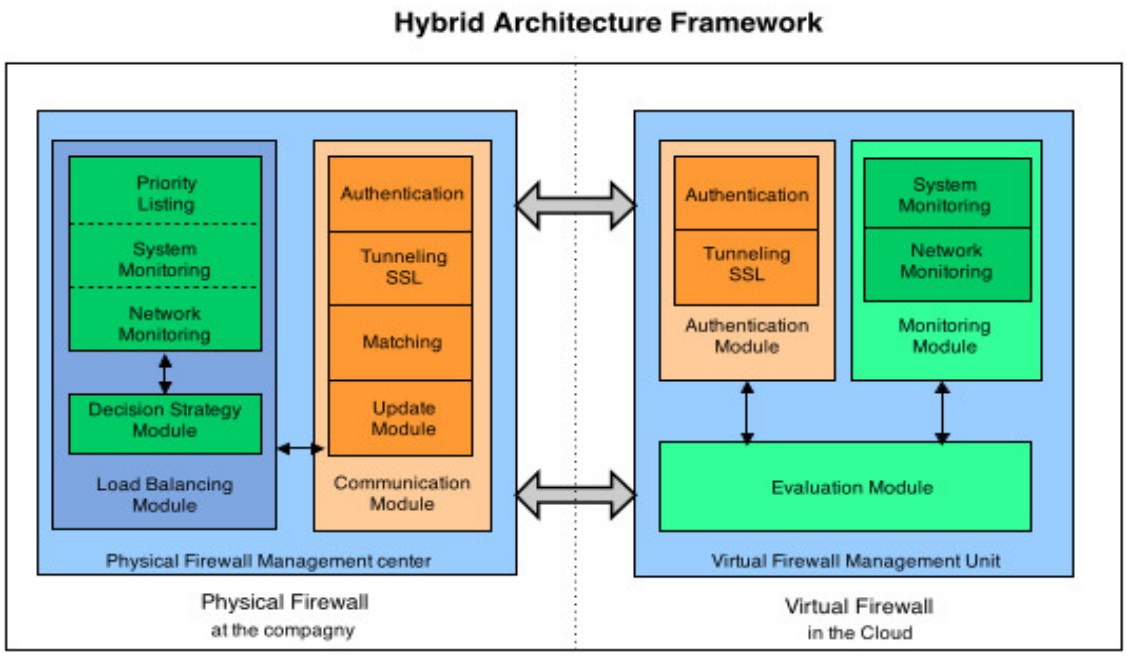

Figure 5 - Structure Logiciel de l'architecture hybride

L'architecture présentée comme hybride est composée de deux parties principales: physique et virtuelle présentées dans la Figure 5. Nous traitons ces deux parties plus loin dans ce chapitre.

Pour assurer un niveau élevé de sécurité et pour éviter les attaques telles que Man in the Middle, le Hijacking et l'usurpation d'identité, nous avons déployé une architecture d'authentification sécurisée, forte et efficace ainsi qu'une solution de gestion de l'identité pour notre service de pare-feu utilisant EAP-TLS technologie à base de cartes à puce.

Plusieurs travaux ont abordé la question de l'authentification et de la gestion des identités et ont proposé des solutions différentes à ce problème. Notre travail s'appuie et s'inspire de la technologie carte à puce, précisément les cartes à puce supportant le protocole EAP-TLS, présenté déjà comme une solution à l'authentification et la gestion des identités dans les articles [49][50]. L'utilisation de la technologie carte à puce offre:

- Système d'identité convergent appelé ``SSL-identity`` qui nécessite une authentification mutuelle entre l'utilisateur et l'authentificateur.
- Un serveur d'authentification supportant la technologie TLS-Tandem qui permet d'utiliser EAP-TLS pour l'authentification et l'exportation des clés de la carte à puce.
- Un fournisseur Open-ID.

Les identités SSL sont stockées de manière sécurisée dans la carte à puce, ce qui permet à l'utilisateur un accès plus facile aux fournisseurs de services. Par conséquent, cela constitue une solution convergente et sécurisée centrée sur l'utilisateur pour la gestion des identités.

La solution d'authentification, présentée par Urien et al. dans [50], introduit un nouveau design pour un serveur RADIUS, couramment utilisé pour l'authentification, et associé à des grilles de cartes à puces EAP-TLS. Le serveur Radius ne traite que les datagrammes RADIUS et effectue une authentification EAP-TLS.

Nous notons que le choix d'EAP-TLS n'était pas arbitraire. En effet, le protocole EAP assure le transport et l'utilisation des paramètres générés par les méthodes EAP et garantissant des mécanismes d'accès au réseau ainsi que l'authentification mutuelle entre le client et le serveur[51]. En outre, le protocole EAP est soutenu par la plupart des serveurs RADIUS, apportant une plus grande complexité de mécanismes d'authentifications traditionnelles.

a. Authentification et gestion des identités :

La mission principale du module d'authentification est d'assurer l'authentification et l'établissement d'un tunnel sécurisé via VPN entre les pare-feu physiques et virtuels. Le module d'authentification offre la possibilité d'utiliser différents protocoles d'authentification, car basé sur un serveur Radius.

Nous déployons un outil open source nommé freeRADIUS qui offre l'authentification des utilisateurs via plusieurs protocoles tels que: PAP, CHAP, MS-CHAP, MS-CHAPv2, SIP Digest, et toutes les méthodes EAP communes. Nous utilisons EAP-TLS basé sur le protocole SSL car la négociation SSL est effectuée sur EAP. Nous concevons et développons une architecture d'authentification cohérente pour le Cloud-based Firewalling service comme le montre la Figure 6. L'architecture proposée assure une gestion efficace des communications sécurisées entre les pare-feu virtuels situés dans le Cloud et le pare-feu traditionnel de l'entreprise.

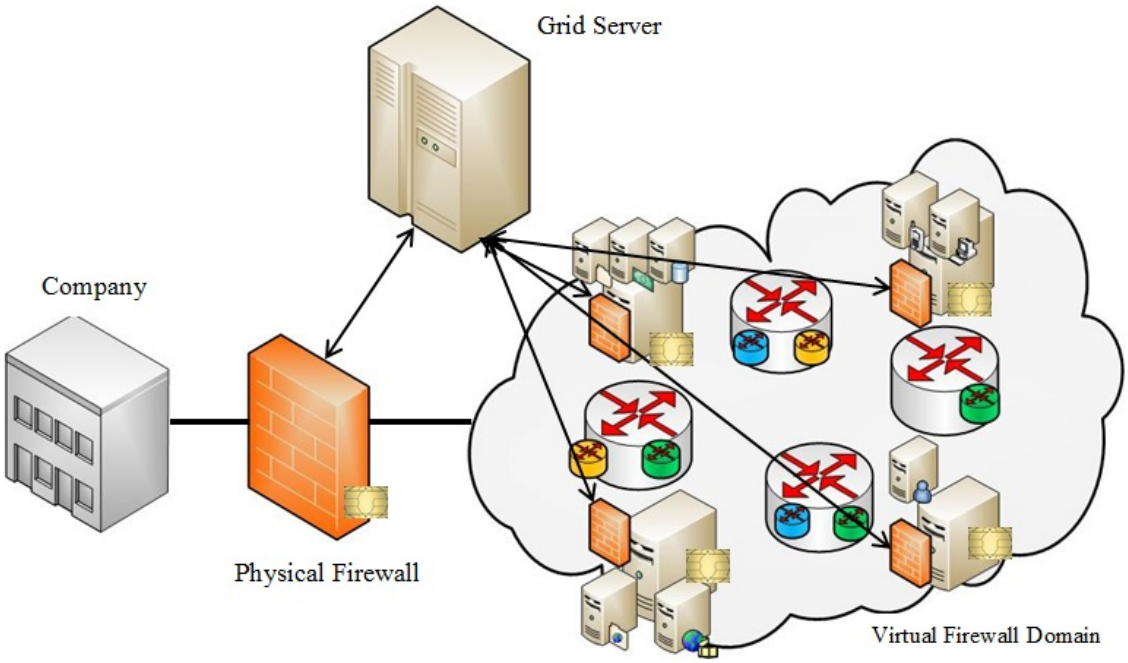

Figure 6 - Architecture d'authentification et de gestion d'identités

Le module d'authentification proposé doit identifier à la fois le pare-feu physique et les pare-feu virtuels correspondants. Il se base sur cinq éléments clés qui sont: Firewall Virtuel noté VF, Grid Server noté GS, Firewall Physique noté PF, Cloud Physical Server[1] noté CS et des certificats. La Figure 7 représente le schéma relationnel des différents acteurs de l'architecture d'authentification.

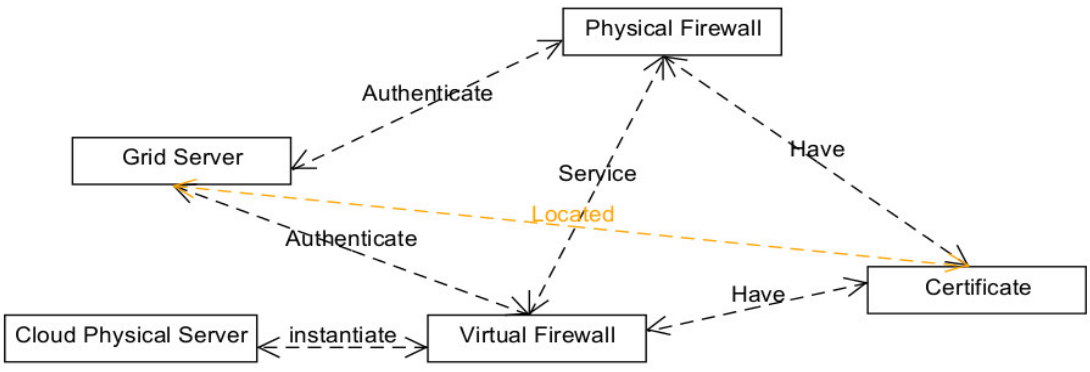

Figure 7- Schéma relationnel de l'architecture d'authentification

[1]Serveur Cloud physique est un serveur physique de grande capacité permettant la virtualisation et l'hébergement ou instanciation de différentes machines virtuelles.

On définit notre module d'authentification comme suit :

- Pare-feu physique représente l'infrastructure de sécurité physique de l'entreprise.
- « A » est un ensemble de PF, chaque entreprise (client) possède un nombre K de pare-feu physiques. $|A| = K$.
- Un pare-feu virtuel feu est une machine virtuelle qui exécute un programme de pare-feu avec des opérations telles que l'analyse, le monitoring, le reporting…etc.
- Chaque pare-feu physique et virtuel possèdent un certificat unique stocké dans sa carte à puce
- VF est instancié au sein du CS, chaque CS peut instancier un nombre 'N' de VF. Par conséquent, chaque CS est équipé avec 'N' lecteurs de cartes à puce.
- Le Grid serveur est un ensemble de cartes à puce disposé en lame qui supporte un nombre total de cartes à puces représentant les pare-feu physiques et virtuels égal à $M + (K * M)$ qui équivaut à:
    - jusqu'à 'M' Firewall physique,
    - $K \times N$ pare-feu virtuels.

Le pare-feu physique établit un tunnel sécurisé à un pare-feu virtuel donné, afin de gérer et de transférer le trafic. Cet accès ne peut être autorisé jusqu'à réussite des étapes d'authentification. Les étapes d'authentification sont divisées en trois grandes étapes telles que présentées dans la Figure 8:

- Authentification du pare-feu physique
- Authentification du pare-feu virtuel
- Mise en place d'un tunnel sécurisé via EAP-TLS entre le pare-feu physiques et les pare-feu virtuels correspondants

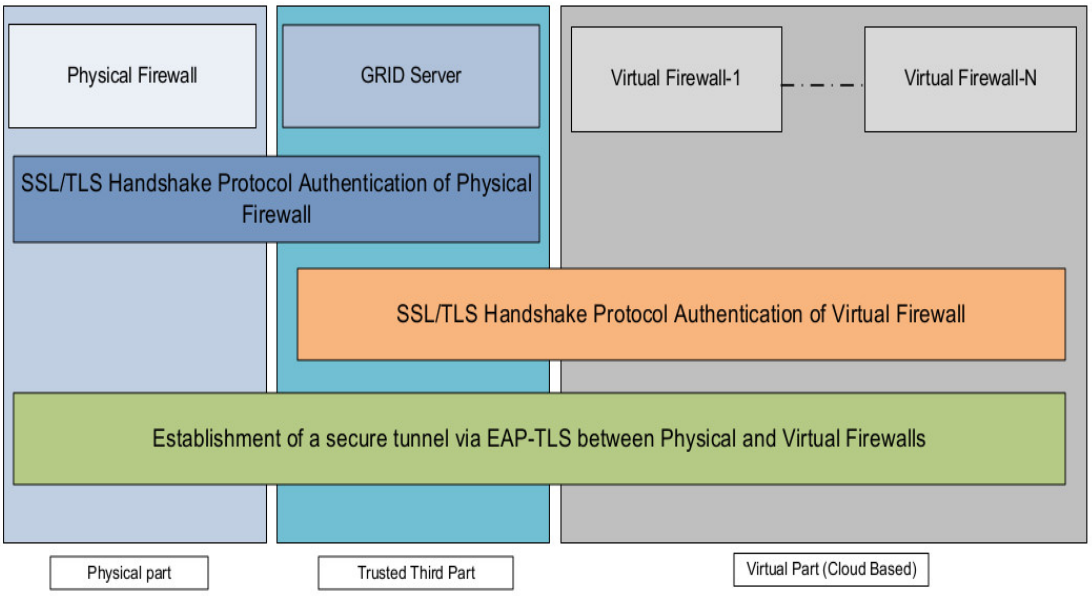

Figure 8 - Etapes de mise en place du tunnel sécurisé

La première étape consiste à authentifier le PF. Le protocole d'échange entre le pare-feu physique et le Grid serveur est basé sur le protocole client/serveur ; ainsi les échanges entre le PF et le GS (Figure 9) sont les suivants :

- ClientHello: Ce message contient: la version du protocole SSL, nombre aléatoire, l'ID de session. Suite de chiffrement: la liste des suites de chiffrement sélectionnées et des algorithmes de décompression.
- ServerHello: Ce message contient les mêmes éléments que le message ClientHello mais correspondants au Grid Server.
- Certificat: Ce message contient le certificat du serveur.
- ServerKeyExchange: contient le certificat de signature
- CertificateRequest: le serveur nécessite un certificat client
- ServerHelloDone: la fin de l'envoi de messages
- ClientKeyExchange: Ce message contient PreMasterSecret chiffré en utilisant l'aide de la clé publique du serveur.
- CertificateVerify: vérification explicite du certificat client.
- Finish: mettre fin au protocole d'établissement de liaison et le début de la transmission de données.

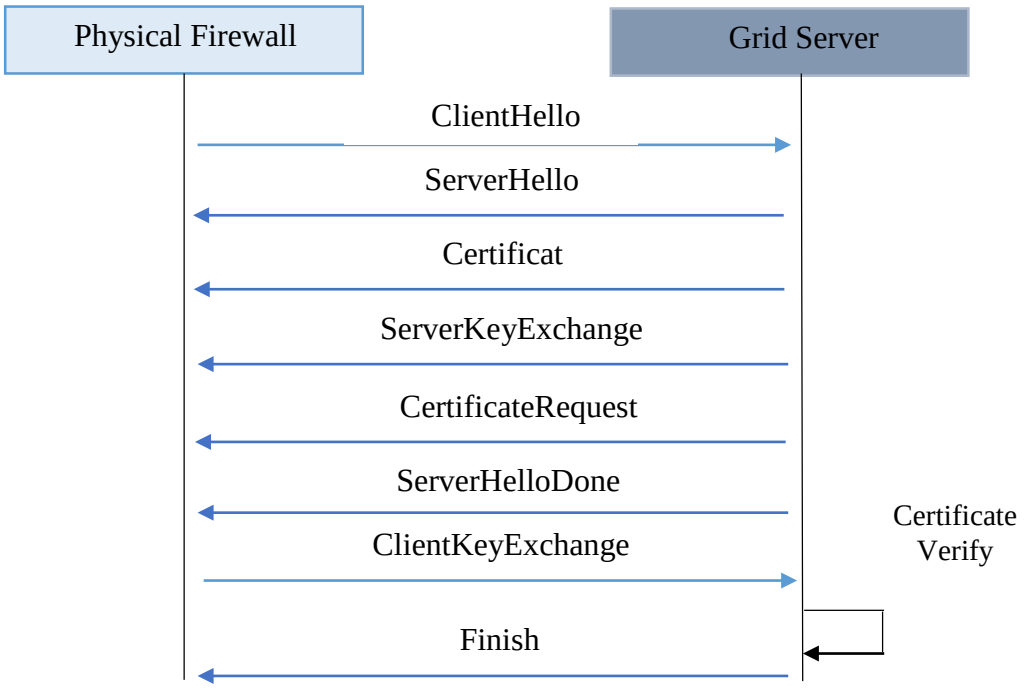

Figure 9- Echanges authentification PF et GS

Le GS doit à présent identifier et authentifier tous les pare-feu virtuels liés au PF authentifiés, tels que développés dans l'étape précédente d'authentification du pare-feu physique : c'est-à-dire que le VF réalise le même schéma d'authentification. Ces étapes sont réalisées par une paire de cartes à puce situées dans GS et CS via le protocole EAP-TLS comme indiqué dans[50].Une fois que tous les firewalls virtuels correspondants au PF sont authentifiés, le GS envoie au pare-feu physique toutes les informations nécessaires sur ses pare-feu virtuels correspondants, permettant ainsi le plein accès à toutes les ressources de pare-feu virtuels.

b. Le Physical Firewall Management Center (PFMC) :

La partie physique présentée dans la Figure 5 est déployée dans le pare-feu physique comme entité logicielle nommée 'Physical Firewall Management Center' ; en réalité c'est un outil de gestion que l'on a développé et mis en place et qui nous permet de fournir une meilleure gestion de l'architecture hybride ainsi que des performances plus élevées. Par la suite nous présenterons les différents modules qui composent le PFMC.

i.Le module de communication :

Le module de communication est responsable du maintien d'un environnement de réseau de confiance ; il est composé de trois sous-modules qui sont: le module d'authentification, Matching et de Mise à jour.

Deux étapes sont nécessaires: l'authentification et l'établissement d'un tunnel sécurisé entre le pare-feu physique et virtuel. Le module d'authentification que nous avons présenté précédemment offre nous le rappelons la possibilité d'utiliser différents protocoles d'authentification.

Le module de **Matching** quant à lui permet de faire correspondre les informations reçues des diverses 'Virtual Firewall Management Unit' à leurs expéditeurs. Ces informations sont envoyées au module de **mise à jour** qui contient toutes les informations relatives aux pare-feu virtuels, il est mis à jour régulièrement pour assurer une bonne compréhension et une vision globale de tout l'environnement de notre architecture hybride et aide le module de stratégie et décision à faire le nécessaire.

ii.Le module de décision et de stratégie :

Le **module de décision et de stratégie** est un module important ; il décide de rediriger ou pas le trafic vers les firewalls virtuels ; il permet aussi la mise en place d'une stratégie d'équilibrage de charge du trafic entre les différents pare-feu virtuels. Pour ce faire, le **module de décision et de stratégie** crée une **liste des priorités** en fonction des informations reçues par **le module de monitoring système et réseau**. Il reçoit des informations telles que la charge du processeur et de la mémoire, la bande passante effective entre le pare-feu physique et virtuel ...etc. Cette liste définit un ordre de priorité des pare-feu virtuels les plus disponibles et associe à chaque pare-feu virtuel un pourcentage du débit que le pare-feu physique lui adressera, tout ceci à la condition que la charge CPU physique soit égale ou supérieure à 40%. C'est donc une règle statique.

iii.Le module de Load-Balancing :

Le module de Load-Balancing reçoit ses ordres du module de décision et de stratégie. La première fonction de ce module est de mettre en place un trafic partagé et donc d'appliquer les règles énoncées par le module de stratégie et de décision. Il fonctionne d'une manière très dynamique, en spécifiant le port, le protocole et l'adresse IP de destination. Il doit impérativement interagir avec le module d'authentification pour obtenir des informations fiables du destinataire comme l'adresse IP et le numéro de port où rediriger le trafic. La deuxième fonction est de basculer (Switcher) le trafic légitime revenant des pare-feu virtuels au réseau local (LAN) de l'entreprise sans aucune procédure d'analyse.

c. Le Virtual Firewall Management Unit (VFMU) :

La partie virtuelle est un ensemble de pare-feu virtuels qui sont des machines virtuelles gérées comme un pare-feu fourni par le fournisseur Cloud. Les principales missions des firewall virtuels sont de fournir le filtrage de paquets et le monitoring à l'aide de logiciels tels que Netfilter[52]. Les firewalls virtuels analysent soigneusement les données provenant du pare-feu physique. Ces données sont analysées sur la base des politiques de sécurité de l'entreprise et redirigent le trafic légitime au réseau local (LAN).

A cet effet, tous les pare-feu virtuels exécutent un 'VFMU' dans le but de faciliter la communication et la gestion des composants des modules de notre architecture hybride. Ainsi, le VFMU est un outil que nous avons développé et qui permet aux pare-feu virtuels d'interagir avec le pare-feu physique correspondant (c'est à dire avec PFMC). Il est composé de trois sous modules: authentification, Monitoring et Evaluation.

i. Le module d'authentification :

Il coopère avec son composant homologue du PFMC afin d'établir un tunnel sécurisé entre le pare-feu virtuel et physique et d'échanger le flux.

ii. Le module de monitoring :

Le module de monitoring est constamment entrain de sonder notre système (pare-feu virtuel) pour obtenir une vue d'ensemble de l'état du réseau et du système. Il ne fournit pas seulement un aperçu du système virtuel, il informe et alerte sur les problèmes potentiels lors de l'exécution. Ainsi, les alertes sont envoyées au **module d'évaluation**.

iii. Le module d'évaluation

La principale fonction du module d'évaluation comme son nom l'indique est d'évaluer l'état du pare-feu virtuel en se basant sur les informations envoyées par le module de monitoring. Il exécute une fonction qui donne en sortie un état du pare-feu virtuel. Elle agrège les différents paramètres du pare-feu virtuel et envoie cette information au module décision et stratégie pour adapter et mettre à jour ses informations.

Pour observer et valider l'efficacité de notre architecture, nous déployons un véritable banc d'essai et proposons deux schémas de déploiement. En plus de cela, deux

cas d'utilisation sont utilisés pour avoir des résultats cohérents avec ce qui se passe en exploitation. La prochaine partie détaille les modèles de déploiement que nous proposons.

2. Modèle de déploiement de l'architecture :

Pour valider notre proposition, nous avons déployé deux différents schémas qui nous permettent d'augmenter la puissance de calcul du pare-feu physique. Nous avons nommé le premier schéma « Secure Forwarding Architecture » (SFA), et le second est nommé Secure Sharing Architecture.

Pour pouvoir comparer les performances des deux schémas proposés, nous avons choisi de mettre en place un schéma de référence basé sur une **architecture à pare-feu unique** comme illustrée dans la Figure 10. Notez que nous utilisons cette architecture de référence car il s'agit d'une topologie de base couramment utilisée par la plupart des petites et moyennes entreprises. Le pare-feu physique reste la principale passerelle de l'entreprise qui connecte entre la LAN et Internet. Ainsi, tout le trafic légitime venant de pare-feu virtuels traverse le pare-feu physique pour atteindre le LAN ou DMZ. Pour rappel, le modèle que nous avons présenté précédemment est conçu pour gérer la coopération entre la partie virtuelle et la partie physique de notre architecture hybride.

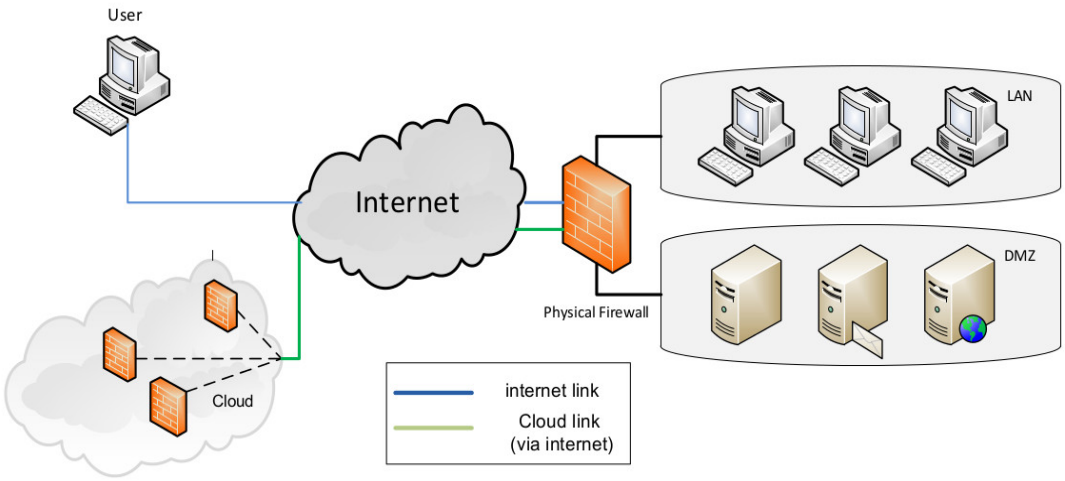

Figure 10- Architecture à pare-feu unique (référence)

a. Secure Forwarding Architecture (SFA) :

L'idée principale de ce modèle de déploiement est de transmettre tout le trafic entrant vers les firewalls virtuels qui prennent en charge l'analyse et l'application des politiques de sécurité de l'entreprise, puis ils retransmettent les paquets légitimes vers

le pare-feu physique. Les firewalls virtuels assurent toutes les fonctions de filtrage. Ainsi, le pare-feu physique devient un simple routeur car il ne fait que router tous les paquets entrants pour être analysés au niveau des pare-feu virtuels. Cette inspection rappelons-le, repose sur la même politique de sécurité qui est appliquée au niveau du pare-feu physique.

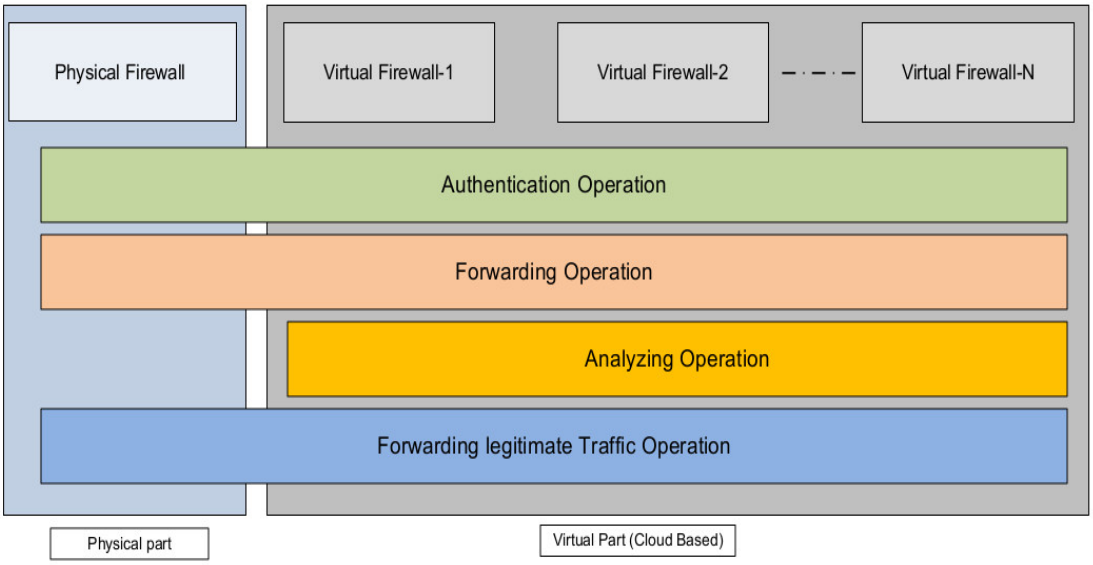

Figure 11- Diagramme de séquence pour Secure Forwarding Architecture (SFA)

Pour manager ce schéma de déploiement, notre architecture hybride gère 4 opérations nécessaires:

- Opération d'authentification : comme présenté auparavant le module d'authentification est exécuté au niveau des deux types de pare-feu. Il authentifie tous les pare-feu et instancie un tunnel sécurisé entre le pare-feu physiques et chaque pare-feu virtuel.
- Opération de Transfert: chaque paquet reçu est transféré au pare-feu virtuel via le tunnel sécurisé. Le choix du pare-feu virtuel est arbitraire.
- Analyse des flux: les pare-feu virtuels interceptent le trafic afin qu'il soit analysé sur la base des politiques de pare-feu physiques.
- Retransmission du trafic légitime: Après l'étape d'analyse, le trafic légitime est envoyé vers le pare-feu physique qui le renvoie au LAN sans analyse.

Pour résumer, l'analyse est totalement déportée au sein du Cloud et des pare-feu virtuels ainsi le pare-feu physique se transforme en simple routeur.

b. Secure Sharing Architecture (SSA) :

Le **Secure Sharing Architecture** possède les mêmes composants logiciels que le **Secure Forwarding Architecture** mais utilisés différemment pour exécuter un schéma différent. En fait, l'analyse est réalisée à la fois par les pare-feu virtuels et physiques en même temps avec différents pourcentages de charge comme le montre la Figure 12. Ainsi, la principale différence entre les deux schémas de déploiement est principalement dans la fonctionnalité d'analyse qui est partagée dans les deux environnements (physique et virtuel).

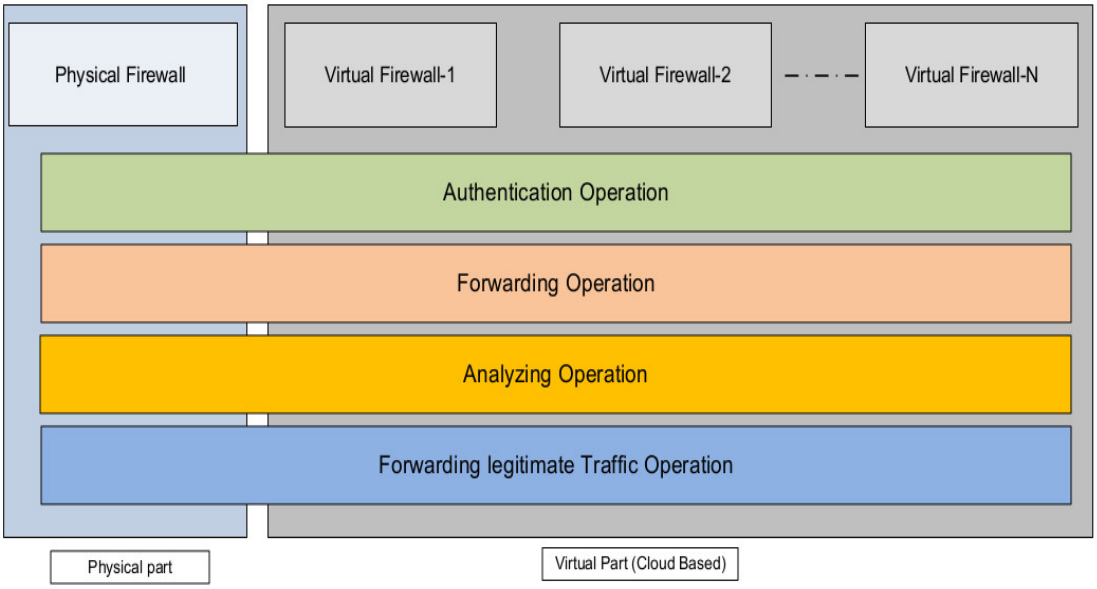

Figure 12- Diagramme de séquence pour Secure Sharing Architecture (SSA)

Nous avons présenté dans cette partie les schémas de déploiement de l'architecture hybride du Cloud-based firewalling services ; la section suivante traite des cas d'utilisation qui nous permettent de valider l'efficacité de notre solution. Nos cas d'utilisation représentent respectivement deux principales problématiques des environnements réseaux, la congestion de la bande passante et déni de service.

3. Cas d'utilisations et implémentation :

Nous voulons démontrer dans cette partie la faisabilité des méthodes de déploiement proposées (SFA et SSA). Pour ce faire nous faisons subir aux schémas de déploiement des cas d'utilisation réels: attaque DDoS et la congestion du réseau. Par la suite, nous comparons les résultats obtenus avec l'approche « de Référence » qui représente le cas où notre architecture n'a pas encore été appliquée.

a. Cas d'utilisations :

i.Attaque de déni de services :

Une attaque Distributed Denial-of-service vise à empêcher les utilisateurs légitimes d'un service d'accéder à ce dernier. La plupart des tentatives des attaquants impliquent de pouvoir utiliser l'ensemble des capacités du réseau et rendre les ressources réseau indisponibles. Nous voulons démontrer que notre architecture permet d'atténuer l'effet de cette attaque. Nous atteignons cet objectif en augmentant la puissance de calcul du pare-feu physique et nous l'adaptons contre l'énorme quantité de trafic généré par les attaques DDoS. Pour réaliser une telle attaque, nous utilisons l'outil Hping [53] pour générer des attaques de types DDos. Cet outil permet de simuler un déni SYN attaque de service. La durée de l'essai est de 120 secondes. Nous répétons l'attaque toutes les 30 secondes.

ii.La congestion réseau :

En règle générale, nous mettons en place un pare-feu de type packet-filter à la frontière des serveurs internes afin de filtrer tout le trafic d'entrée. Il représente un point d'entrée unique. Les problèmes liés à la capacité du processeur et la bande passante du firewall peuvent entrainer immédiatement un cas de congestion (Bottleneck) du réseau. Donc, un défi important auquel notre architecture doit faire face est la congestion de la bande passante réseau causée par le goulot d'étranglement (Stress point) que représente le firewall physique. En appliquant nos méthodes (SSA et SFA), nous démontrons que ce problème est presque exclu.

b. Environnement de simulation :

Nous avons déployé un banc d'essai au niveau du laboratoire composé d'un ensemble d'équipements réseaux, comme illustré dans la Figure 10 représentant notre architecture de référence. L'environnement de simulation est composé d'un pare-feu physique exécutant un logiciel nommé NetFilter [52]qui assure la fonction de filtrage séparant les serveurs Web internes situés dans différentes zones démilitarisées(DMZ) de l'internet. Nous créons des pare-feu virtuels installés sur un nuage local. Chaque pare-feu virtuel est une machine virtuelle équipée d'un processeur simple, Intel(R) Xeon(R) CPUE5335. Les processeurs Intel fonctionnent à 2.00GHz, avec512Kode RAM. Le système d'exploitation est Ubuntu 12.04(x86-64-linux-gnu). Afin d'assurer la fonction de filtrage, nous exécuterons des instances Netfilter au niveau des firewalls virtuels avec les mêmes règles que la passerelle pare-feu physique.

Les mesures de performance sont prises en utilisant l'outil Iperf [54] .Cet outil est basé sur une architecture client/serveur et enregistre les performances réseau en mesurant plusieurs paramètres QoS. Nous nous concentrons en particulier sur trois paramètres essentiels: CPU, la latence et le taux de perte des paquets. Ces paramètres sont étudiés et comparés dans nos cas d'utilisations.

c. Résultats et discussion :

Nous présentons et discutons les résultats de nos différentes méthodes (de base, SFA et SSA) dans deux cas d'utilisations présentés ci-dessus: les attaques DDOS et la congestion du réseau.

i.Attaque de déni de service :

La Figure 13 présente les mesures de la charge CPU du pare-feu physique dans le cas d'une attaque DDOS périodique. L'axe des X indique le temps en secondes. L'axe des Y indique le pourcentage d'utilisation du processeur. Nous analysons les résultats dans chaque intervalle de 30 secondes.

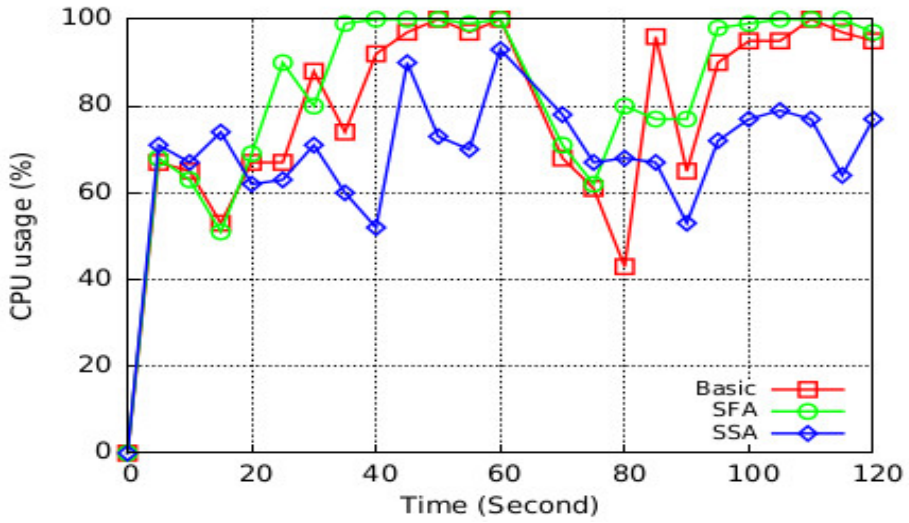

Figure 13- Charge CPU sous attaque DDOS

- Absence d'attaque DDoS [0s :30s] : Dans cet intervalle, l'attaque DDos n'est pas encore exécutée. Nous observons que la consommation CPU du pare-feu physique dans le cas de SFA est approximativement similaire à celle de l'architecture de référence. La méthode SSA consomme plus de CPU parce que le pare-feu physique procède à des tâches d'équilibrage de charges supplémentaires.
- Présence d'attaque DDoS [30s : 60s] : Nous remarquons clairement une consommation plus élevée du CPU due à l'attaque DDOS. Cependant, la

méthode SSA consomme moins de CPU par rapport aux autres méthodes (référence et SFA).

- Absence d'attaque DDoS [60s : 90s] : Nous pouvons constater que la consommation CPU est en diminution dans cet intervalle. Nous notons que la méthode SSA est encore la plus basse consommation CPU par rapport aux autres méthodes.
- Présence d'attaque DDoS [90s : 120s] : le DDOS est appliqué à nouveau pendant 30 secondes. Nous nous apercevons que le pare-feu physique atteint rapidement la capacité maximale du processeur (100%) dans le cas de référence et du schéma SFA. Alors que dans l'approche SSA, nous remarquons une petite augmentation de l'utilisation du processeur qui ne dépasse pas80%.

Nous concluons dans la Figure 13 que la consommation CPU du schéma n'atteint jamais la capacité maximale du pare-feu physique, et ce même lorsque nous avons une attaque DDOS. Cela prouve que le partage de charge entre le pare-feu physique et virtuel est la solution adéquate qui permet de faire face à l'attaque DDOS. Il améliore la consommation des ressources du pare-feu physique et réduit la probabilité de goulot d'étranglement.

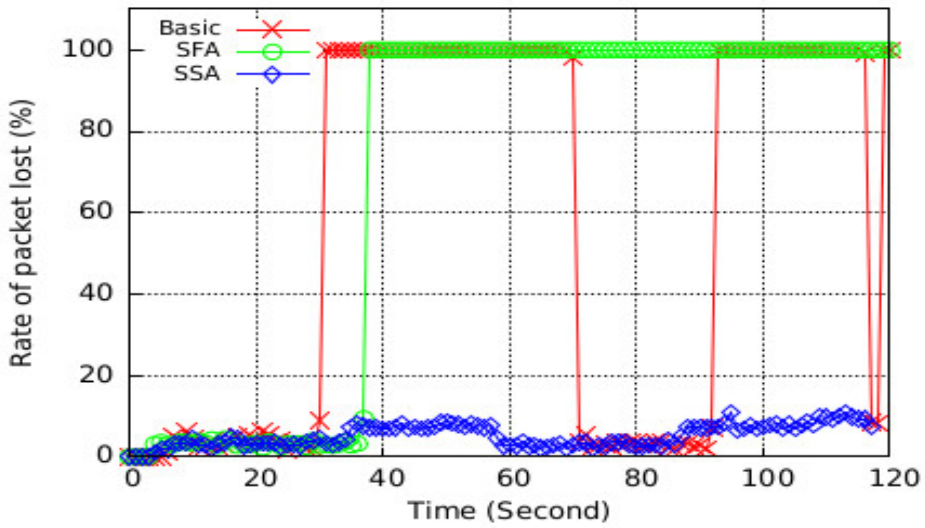

Figure 14- Taux de pertes avec attaque DDoS

De la même façon, nous étudions si nos méthodes aident à réduire le taux de perte de paquets au cours d'une attaque DDoS. La Figure 14montre le taux de perte de paquets dans chacun des schémas de déploiement (SFA, SSA, Référence) lors de l'attaque DDoS. Nous observons que le taux de perte commence à monter après 35

secondes car l'attaque DDos est déclenchée à 30 secondes et progressivement elle impacte l'architecture.

Les deux méthodes SFA et de référence traitent tout le trafic d'entrée que cela soit pour de l'analyse ou pour du routage ; les taux de perte augmentent rapidement et après 38 secondes on atteint les 100%. Cela signifie que le pare-feu physique est « **Down** » car toutes ses ressources sont consommées. Par contre, en ce qui concerne le schéma SSA, le pourcentage de taux de perte de paquets est inclus dans une plage de 0% à 10% sous l'attaque DDoS.

Cela démontre que la méthode SSA est une solution potentielle pour faire face à l'attaque DDoS. Comme les opérations d'analyse du trafic sont partagées entre le pare-feu physique et virtuel, les ressources du pare-feu physique ne sont pas totalement saturées. Nous concluons que le schéma de déploiement SSA est une solution intéressante pour faire face aux effets des attaques de type DDoS.

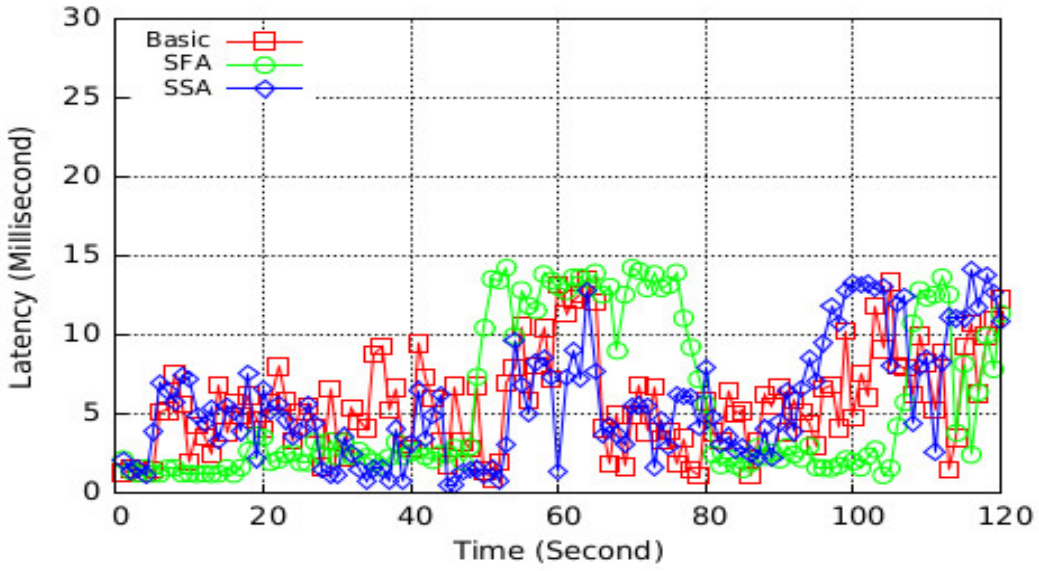

Figure 15- Latence avec attaque DDoS

Nous nous intéressons à la latence du réseau lors de l'utilisation de nos méthodes SFA, SSA et Référence. La Figure 15 montre que l'utilisation de notre architecture hybride améliore les performances réseaux de manière significative, principalement le paramètre latence par rapport à notre architecture de base. Nous observons qu'après 45 secondes la latence est augmentée en raison de l'attaque DDoS qui est déclenchée à partir de 30 secondes. La latence dans le cas SFA augmente à un rythme plus lent par rapport aux méthodes de référence et SSA car le pare-feu physique exécute

exclusivement le routage des paquets et donc consomme moins de temps pour traiter un paquet, tandis que le filtrage et le contrôle des paquets sont effectués par les pare-feu virtuels.

Dans les résultats précédents, nous avons constaté que le schéma SSA améliore l'utilisation du processeur et le taux de perte de paquets dans le cas d'une attaque DDoS. Cependant, nous observons que le temps de latence est supérieur à la méthode SFA. Par conséquent, l'utilisation du schéma SSA est un compromis à faire entre une latence un peu plus importante et des taux de pertes acceptables.

ii.La congestion de réseau :

Nous allons démontrer comment notre proposition d'architecture hybride permet d'éliminer tout effet de goulot d'étranglement au niveau du pare-feu physique. Nous augmentons le trafic généré pour saturer la bande passante du pare-feu physique jusqu'à atteindre les 100% de saturation. On augmente à chaque fois de 10% le trafic et on enregistre les différents paramètres qui nous intéressent (CPU, latence). La Figure 16 correspond à la charge du processeur du pare-feu physique en variant avec le pourcentage de saturation de la bande passante.

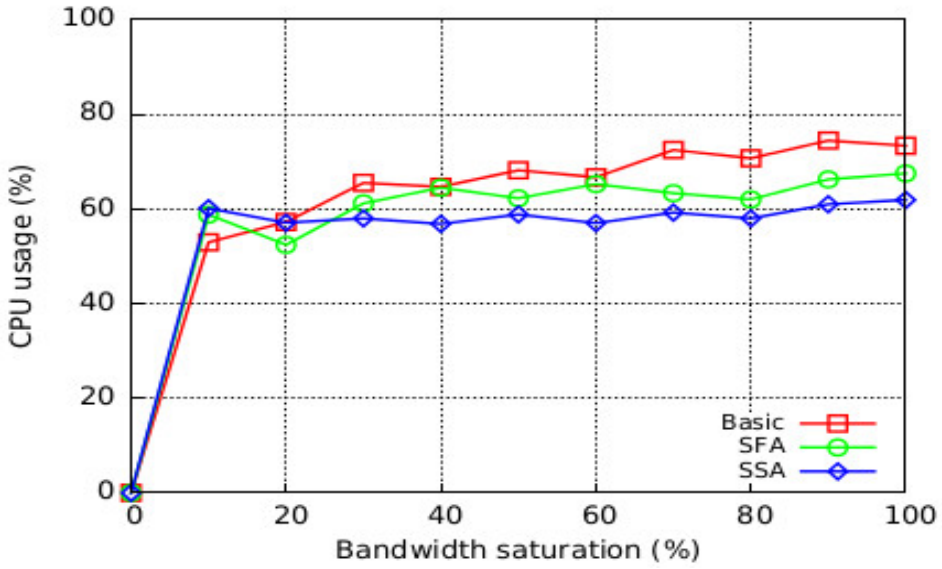

Figure 16- Charge CPU avec Congestion réseau

Nous notons que dans toutes les méthodes (SSA, SFA et de Référence), l'utilisation du processeur ne cesse de croître car la consommation de bande passante augmente (le nombre de paquets à traiter est en augmentation aussi). Nous observons que la méthode SSA offre un gain de plus de 10 % par rapport à SFA et l'architecture

de référence. En conséquence, avec cette méthode, on peut augmenter l'utilisation et l'efficacité du processeur du pare-feu physique.

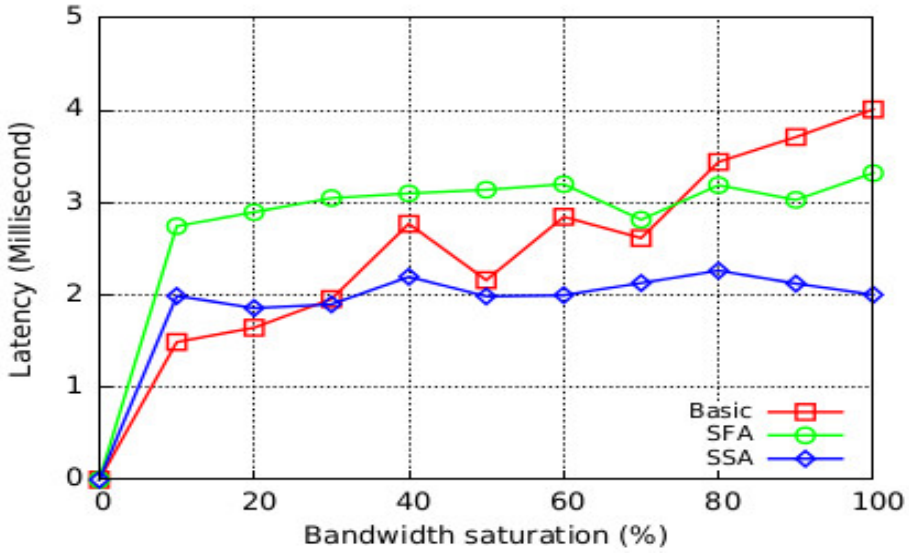

Figure 17- Latence avec Congestion réseau

Dans la Figure 17, l'axe des Y représente la mesure de la latence du réseau (RTT: Round-trip Delay) et l'axe des X représente le pourcentage de saturation de la bande passante. Nous notons que la méthode SSA améliore considérablement le temps de la latence du réseau en comparant avec la SFA et l'approche de référence. Le temps RTT dépasse légèrement les 2 ms, même lorsque la bande passante est entièrement utilisée.

4. Conclusion

Dans ce chapitre, nous avons présenté une architecture hybride pour du Cloud-Based Firewalling service. Cette architecture a pour but principal d'augmenter la puissance de calcul des pare-feu physiques, avec un faible coût financier, en utilisant les vastes ressources offertes par la technologie Cloud.

Cette architecture est proposée avec deux schémas de déploiement « Secure Forwarding Architecture » et « Secure Sharing Architecture ». Pour démontrer la faisabilité des méthodes de déploiement proposées (SFA et SSA), nous les testons avec deux cas d'utilisations réels qui sont: l'attaque DDoS et la congestion du réseau.

Nous avons comparé les résultats avec une "Architecture de référence". Les résultats présentent une amélioration significative des performances systèmes et réseaux. En fait, l'architecture peut faire face à des attaques distribuées de déni de service et peut être déployée comme un service anti-DDoS.

Cependant et même si notre architecture a montré de bonnes performances, on remarque plusieurs points faibles, comme le fait que l'architecture repose principalement sur le pare-feu physique et que celui-ci devient un point de faille de notre architecture ce qui pourrait nous ramener au point de départ. La gestion de toute l'architecture est statique et ne prend pas rapidement en charge les modifications de charge du trafic. Nous répondrons à ces différents points de faiblesse dans notre prochain chapitre.

IV. Architecture de Management de la Sécurité pour un Service de Firewalling Cloud Based:

Le Cloud Computing et les environnements virtuels en général sont d'importants axes de recherches ces dernières années. La gestion des services de sécurité que propose le Cloud est un enjeu important pour la confiance numérique et la sécurité des environnements virtuels.

Avec la croissance impressionnante du trafic Internet, les utilisateurs (particuliers, entreprises ou gouvernements) font face à des attaques de plus en plus importantes qui engendrent de lourdes conséquences. En effet, avec la sophistication, la complexité et l'imprévisibilité des attaques, il est de plus en plus difficile de se défendre. La plus importante de ces menaces est sans doute le déni de service (DDoS). D'après Yu et al.[4], la question essentielle qui se pose dans une relation Attaque/Défense pour une attaque DDoS est « qui aura le plus de ressources ? ». En fait, c'est une bataille des ressources entre le défenseur et l'attaquant ; si un attaquant possède des ressources suffisantes pour réaliser son attaque, le défenseur ne pourra pas faire face, et vice versa.

De cette constatation se pose la problématique de la limitation de ressources des équipements physiques de sécurité comme les Firewalls. Aujourd'hui, un nouveau type de services de Cloud Computing émerge. De nombreux éditeurs de solutions de sécurité tirent systématiquement parti des modèles du Cloud pour offrir des solutions de sécurité à leurs clients, cela dans le but de remédier à ces problématiques de ressources. En effet, la technologie Cloud repose sur le partage des ressources, il offre une grande quantité de ressources, une haute disponibilité et de grandes performances. La réponse de Yu et al. [4] est basée sur l'utilisation de cette quantité de ressources pour faire face à des attaques de type DDoS et ouvre la voie pour proposer de nouveaux services de sécurité basés sur le modèle « Security as a Service ».

Notre travail est de proposer un service de Firewalling totalement Cloud-Based comme le montre la Figure 18, qui prend en compte le management de toutes les opérations réseaux et sécurité du service. Cette architecture possède les mêmes caractéristiques que le Cloud, flexibilité et disponibilité.

Ce service proposé par un Cloud Provider vient en renfort aux pare-feu traditionnels et aux solutions de détection d'intrusion qui leurs permettra de faire face

aux volumes importants de trafic. Ainsi, cette architecture fournit les ressources nécessaires aux utilisateurs pour faire face à des attaques DDoS mais aussi pour gérer les performances et la fiabilité du service selon leurs besoins. Ceci permet de rapprocher l'expérience du Cloud Provider avec l'expertise métier de l'utilisateur.

Un Cloud Provider offre un service de Firewalling pour ses clients. Celui-ci leur demande de s'abonner à l'offre en leur garantissant le traitement (analyse) d'une capacité de bande-passante de trafic avec des règles de filtrages fonctionnelles et d'autres proposées par l'abonné. Par la suite, le Cloud Provider joue le rôle d'intermédiaire pour le client et ne lui transmet que le trafic légitime, ce qui permet à l'utilisateur d'avoir une capacité de traitement supplémentaire sans avoir à acheter, déployer et maintenir de nouveaux équipements.

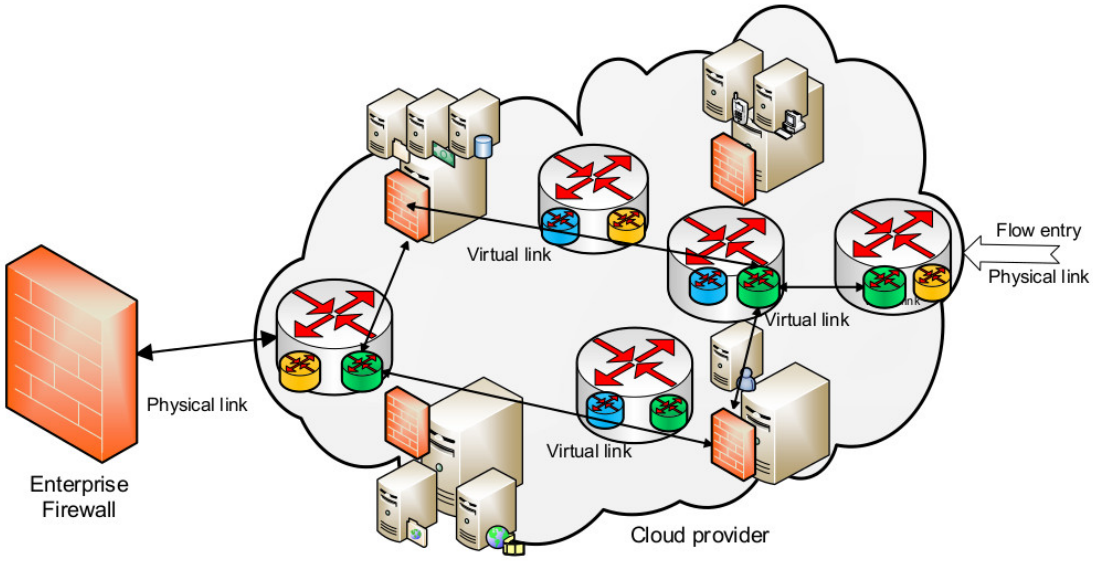

Figure 18- Cloud Based Firewalling service

1. Architecture générale du Firewalling Cloud Based Service :

Le modèle général proposé ressemble à un immense Proxy ou serveur mandataire constitué de trois principaux composants (Figure 19) qui sont : la Front Gateway, Back Gateway et les firewalls virtuels. L'orchestration et la gestion des opérations de ces différents composants représentent notre but principal.

La Front Gateway représente le point d'entrée de notre architecture ; elle garantit la gestion et le traitement nécessaire pour distribuer le trafic sur les instances de pare-feu virtuels. Chaque pare-feu virtuel exécute une application de Firewalling, avec les

opérations traditionnelles d'analyse, de surveillance et de journalisation. Le pare-feu virtuel transmet par la suite le trafic légitime à la Back-Gateway. Cette dernière a pour fonctionnalité principale d'être l'intermédiaire entre le client et le service. Elle recueille le trafic légitime émanant de toutes les instances de firewalls virtuelles et le transmet à sa destination finale. Dans le sens inverse, elle s'occupe de gérer les connections sortantes du client et doit les synchroniser avec le Front Gateway.

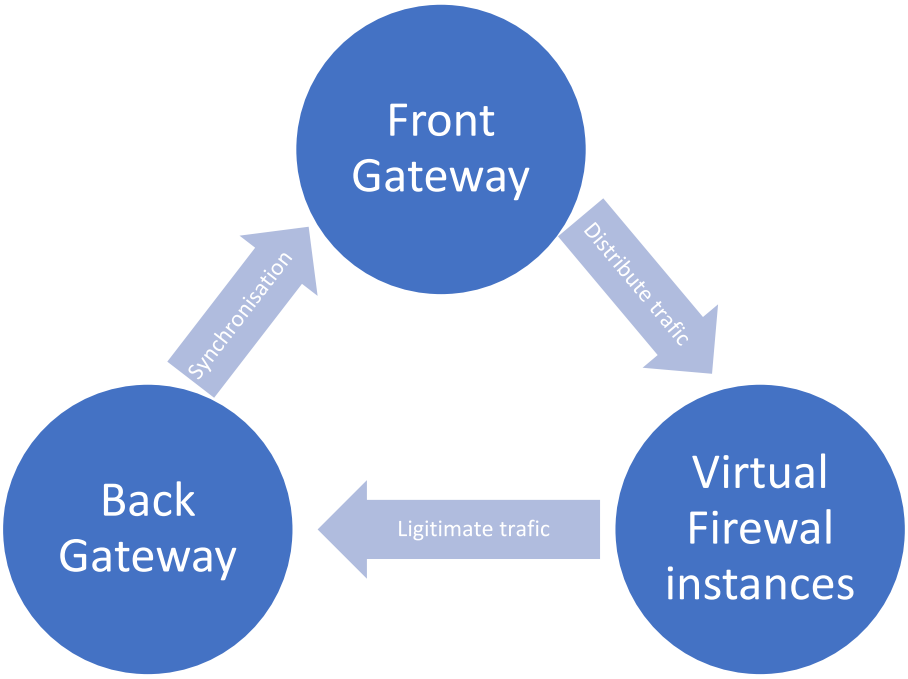

Figure 19- Modèle générale de l'architecture

a. Front Gateway :

La Front Gateway est représentée par un routeur virtuel de grande capacité qui prend en charge une capacité de trafic préalablement connue par le Cloud Provider. Elle a pour principales missions :

- Distribuer le trafic en entrée sur les différentes instances de Firewalls virtuelles.
- Authentifier tous les pare-feu virtuels et créer un tunnel de communication sécurisé.
- Administrer et mettre à jour sa propre base de données des pare-feu virtuels qui lui sont alloués.
- Exécuter les stratégies proposées par le module de décision quant à la distribution du trafic.

Pour chacune de ces missions un module logiciel a été développé comme le montre la Figure 20 :

- Le module de décision manage toutes les opérations réseaux qui concernent le Load-Balancing du trafic aux différentes instances de pare-feu virtuels ; cela englobe le choix de l'algorithme d'équilibrage de charge, le nombre d'instances nécessaires en termes de capacité de bandes passantes à traiter ainsi que la distribution des politiques de sécurité.
- Le module d'équilibrage de charge (Load-Balancing) reçoit ces ordres du module de décision et les met en application au niveau réseau. En pratique, il met en place des règles pour pouvoir partager le trafic sur les différentes instances de pare-feu selon les paramètres choisis par le module de décision.
- Base de données des Firewalls virtuels : les informations sur les instances de pare-feu telles que leurs capacités, leurs adresses IP et leurs disponibilités sont stockées et mises à jour au niveau de la base de données d'informations.
- Le module d'authentification permet quant à lui d'authentifier les instances de firewalls mais aussi l'administrateur du service.
- Le module de configuration est là pour aider à l'administration de la solution. En effet, il permet l'ajout de nouvelles informations pare-feu, de modifier l'algorithme d'équilibrage de charge ainsi que les paramètres de décision en fonction du type de trafic reçu.

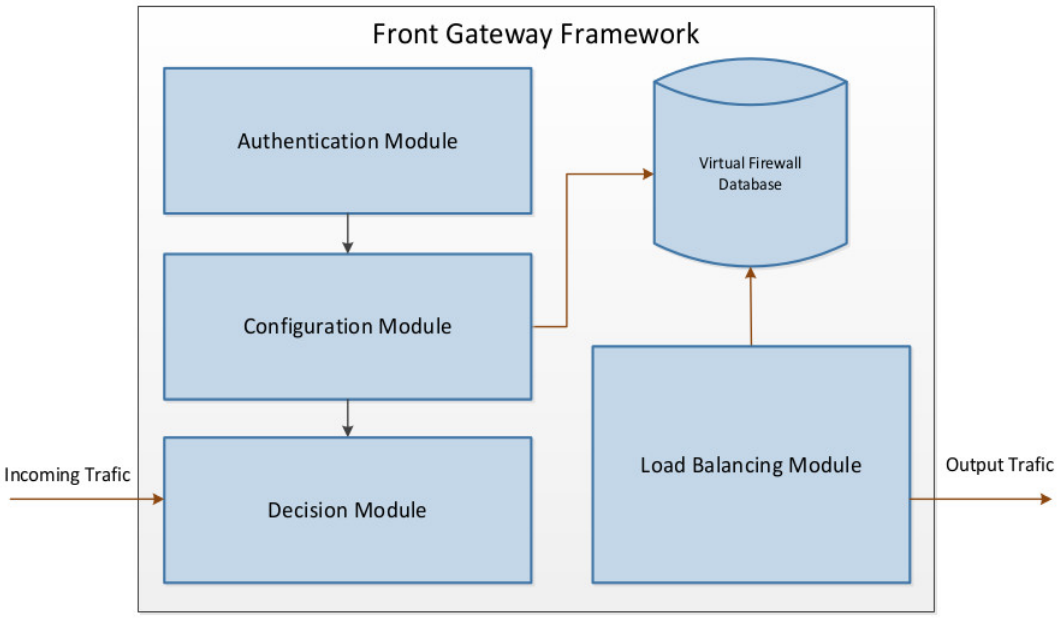

Figure 20- Front Gateway Framework

b. Firewall virtuel :

Chaque instance de pare-feu virtuel joue le rôle d'un pare-feu classique ; tout le trafic entrant doit être vérifié et analysé par le module de Firewalling (Figure 21). La politique de sécurité peut être modifiée à tout moment en fonction des besoins et pour chaque instance par l'administrateur de la solution. Ce dernier s'authentifie au moyen du module d'authentification, qui possède comme fonctionnalité supplémentaire d'authentifier l'instance auprès de la Front-Gateway et de la Back-Gateway. Il est à noter que chaque instance ne prend en charge qu'une partie du trafic global avec un pourcentage décidé par la Front-Gateway. Cependant, la répartition du trafic respecte l'intégrité de la session de façon à envoyer tout le trafic d'une même session, via la même instance virtuelle, cela est nécessaire pour appliquer une inspection de type Stateful.

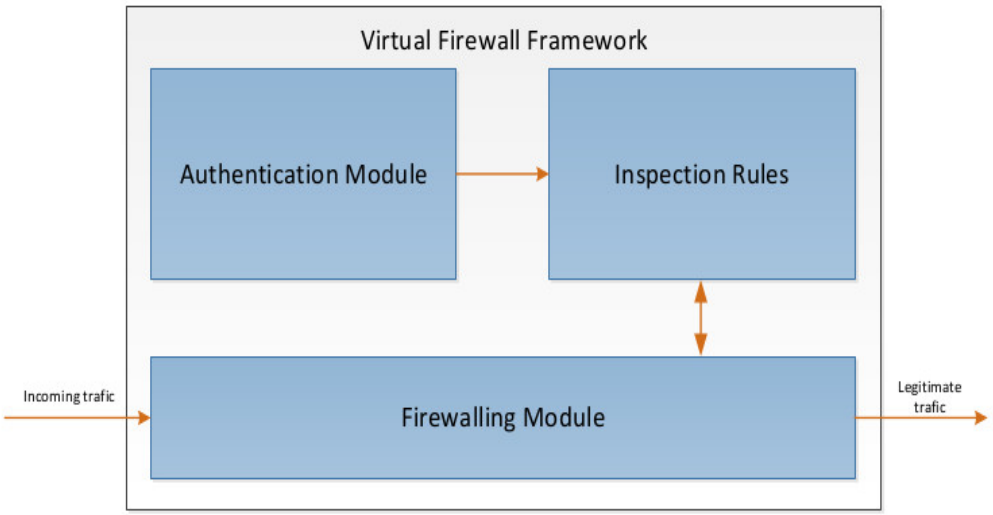

Figure 21- Virtual Firewall Framework

c. Back-Gateway :

Cette passerelle a toute son importance. En effet, elle est le principal et unique lien entre le réseau de l'utilisateur et la solution proposée. C'est un proxy qui relaye les requêtes du client vers la Front-Gateway et qui renvoie le trafic légitime vers le client. Elle assure donc essentiellement deux fonctions (Figure 22) qui sont le réassemblage et la vérification de tous les paquets reçus des différents pare-feu virtuels. En effet, la passerelle est connectée à toutes les instances virtuelles qui lui sont propres ; elle reçoit

le trafic considéré comme légitime. Une fois collecté, le trafic est contrôlé pour la dernière fois avant d'être livré. Cette étape génère également des informations qui sont renvoyées à la Front Gateway à des fins de synchronisation.

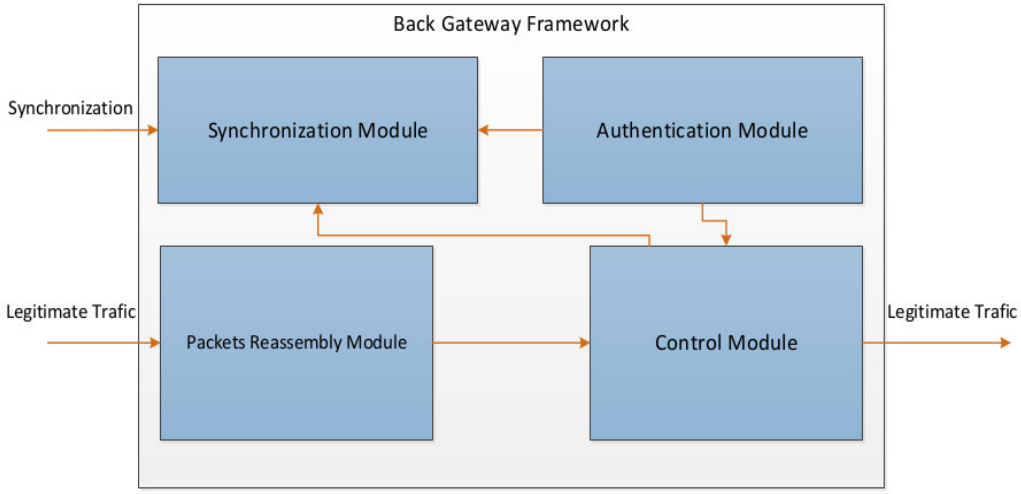

Figure 22- Back Gateway Framework

2. Déploiement et spécification de l'architecture:

L'un des principaux objectifs de l'architecture proposée est d'améliorer les capacités de traitement et d'analyse du trafic d'un utilisateur (particulier, entreprise, gouvernement…etc.) limitées par leurs ressources physiques propres en utilisant de manière optimale et avec des mécanismes de gestion les ressources offertes par le Cloud. Pour ce faire, nous avons développé et déployé notre architecture au sein d'un Cloud local au niveau du laboratoire.

Comme la Figure 18 le montre, la pierre angulaire de notre architecture est la Front Gateway ; celle-ci est représentée par un routeur virtuel de grande capacité sur lequel nous avons déployé un logiciel d'équilibrage de charge. Ce dernier a pour mission d'exécuter la stratégie consentie par le module de décision. Par conséquent, le pourcentage de trafic perçu par chaque pare-feu virtuel est calculé par un algorithme simplifié qui prend en entrée des paramètres prédéfinis et mis à jours dans la base de données des instances. Ceci implique qu'à ce niveau, la solution est statique et ne prend en compte des modifications que si elles sont faites par l'administrateur. Par contre, la Front Gateway prend compte du taux de trafic précédemment envoyé à une instance

pour des décisions futures. Il est à noter que le nombre d'instances des firewalls est connu au départ et provisionné par le Cloud Provider.

Dans le cadre de ce travail, nous n'avons pas pris en compte la gestion de la distribution des règles de filtrage. En effet, toutes les instances de firewalls appliquent la même politique de sécurité avec les mêmes règles de filtrage.

Comme le montre la Figure 23, la solution doit exécuter et gérer différents types d'opérations réseau. L'authentification et le Forwarding qui sont initiés et localisés au niveau de la Front-Gateway, l'analyse qui est principalement et exclusivement effectuée par les instances virtuelles de firewalls, la collecte, l'acheminement et la synchronisation du trafic effectué par la Back-Gateway. Ces opérations sont importantes car elles permettent d'éviter des liaisons directes avec le réseau du client.

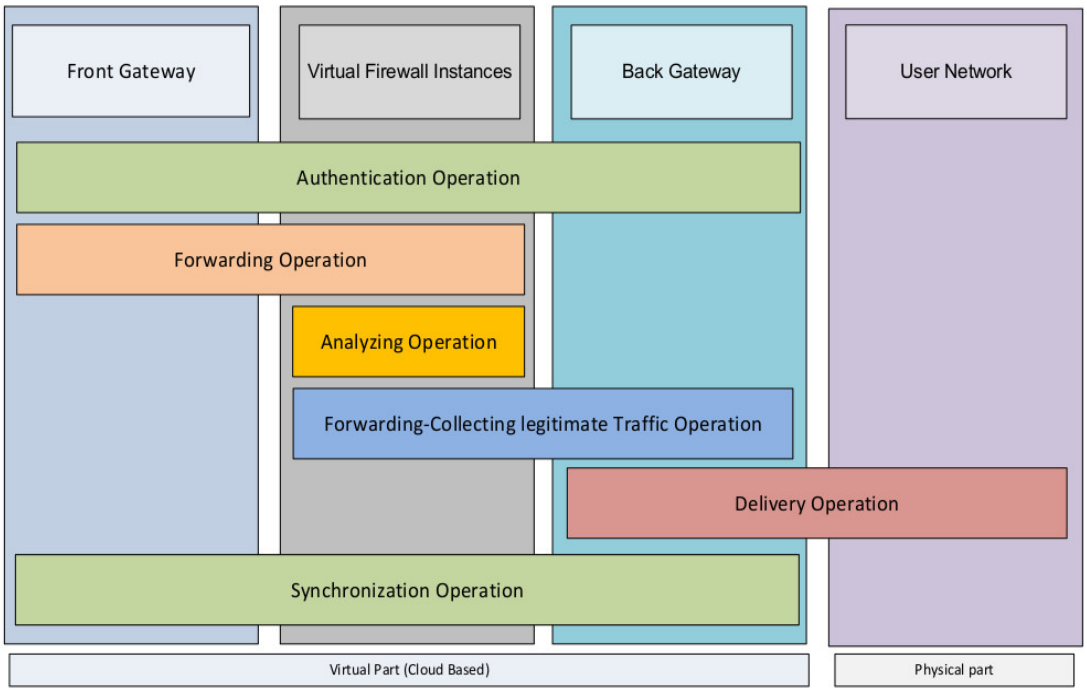

Figure 23- Network operations Framework

3. Opérations réseaux :

Nous présentons dans cette partie les fonctionnalités (opérations) prises en charge par l'architecture :

- L'Authentification : La mission principale du module d'authentification est d'assurer l'authentification et l'établissement d'un tunnel sécurisé via VPN entre les différents acteurs de la solution (Front-Gateway, les instances, la Back-

Gateway et le client). Cette mission est assurée par le module d'authentification, ce dernier offre la possibilité d'utiliser différents protocoles d'authentification, car basé sur un serveur Radius.

- Le Forwarding : l'objectif principal de ce mécanisme est d'arriver à atteindre une fluidité de trafic à travers toutes nos instances ;ce qui permet à toutes les machines de rester disponibles tout en recevant le trafic qu'elles sont en capacité de traiter. En se basant sur les travaux de Khiyaita et al.[55], nous décidons d'adopter une approche centralisée dans laquelle, un seul nœud est responsable de la gestion de la distribution du trafic sur l'ensemble du système.
- L'analyse et Monitoring : Afin d'assurer la fonction de filtrage et d'après nos besoins, nous avons utilisé un pare-feu de type « Stateful ». En effet, ce type de pare-feu nous permet de faire du filtrage applicatif sans dépasser la couche de transport/session. Notre choix est fondé sur la nécessité de garder l'historique des connections. Dans ce but, chaque instance de firewall exécute le logiciel Netfilter [52] avec les mêmes règles.
- La Collecte: comme son nom l'indique représente la collecte des flux des différents pare-feu virtuels vers la Back-Gateway comme point unique de récolte ce qui simplifie les opérations de routage.
- L'acheminement représente l'opération finale du traitement du trafic. En effet, la Back-Gateway transmet le trafic légitime au routeur d'accès du client.
- La synchronisation: Cette opération permet deux choses : relayer les requêtes sortantes du LAN du client pour qu'elles soient prises en compte par la Front-Gateway et les instances de Firewalls. Prendre en charge les messages échangés entre le Front et la Back Gateway pour assurer la collaboration et respecter des contraintes de sécurité.

Le tableau suivant résume les relations d'intermédiation de la Back-Gateway :

Les opérations	Relations d'intermédiation
Collecte	Instances de pare-feu / Back-Gateway
Acheminement	Back-Gateway / Client final
Synchronisation	Back-Gateway / Client final
	Back-Gateway / Front Gateway

Tableau 1 - Tableau des intéractions de la Back-gateway

Nous avons présenté l'architecture générale du Firewalling Cloud Based Service qui prend en compte le management de toutes les opérations réseaux et sécurité du service. La section suivante traite des cas d'utilisation qui nous permettent de valider l'efficacité de notre solution. Nos cas d'utilisation comme précédemment présentés dans la partie a, représentent respectivement deux principales problématiques des environnements réseaux: la congestion de la bande passante et déni de service.

4. Cas d'utilisations et implémentation:

Nous voulons démontrer dans cette partie l'efficacité du service proposé. Pour ce faire nous faisons subir au schéma de déploiement des cas d'utilisation réels: attaque DDoS et la congestion du réseau. Par suite, nous comparons les résultats obtenus avec l'approche « de Référence » qui représentent le cas où notre architecture n'a pas encore été appliquée, en suivant la même procédure que le chapitre 2-3.

a. Cas d'utilisations (Rappel) :

i. Attaque de déni de services :

Une attaque Distributed Denial-of-service vise à empêcher les utilisateurs légitimes d'un service d'accéder à ce service. La plupart des tentatives des attaquants impliquent de pouvoir utiliser l'ensemble des capacités du réseau et rendre les ressources réseau indisponibles. Nous voulons démontrer que notre architecture permet d'atténuer l'effet de cette attaque. Pour réaliser une telle attaque, nous utilisons l'outil Hping [53] pour générer des attaques de types DDos. Cet outil permet de simuler un déni SYN attaque de service. La durée du test est de 120 secondes. Nous répétons l'attaque toutes les 30 secondes.

ii. La congestion réseau :

Un défi important auquel notre service doit faire face est la congestion de la bande passante réseau causée par les différents goulots d'étranglement (Stress point) que représenteraient les instances de firewalls, la Front Gateway...etc. On augmente pour cela à chaque fois de 10% le trafic généré par le logiciel Iperf [54] et on enregistre les différents paramètres réseaux comme la latence et le taux de pertes.

b. Environnement de simulation :

Nous avons déployé un banc d'essai réel au niveau du laboratoire composé d'un ensemble de machines virtuelles déployées dans un Cloud local, comme illustré dans la Figure 18 qui représente notre architecture de référence. L'environnement de simulation

est composé d'un routeur virtuel qui assure les fonctions de la Front-Gateway. Nous créons des instances de pare-feu virtuels installés et déployés en parallèle. Nous avons exécuté nos tests avec respectivement 2 et 3 instances. Chaque pare-feu virtuel est une machine virtuelle équipée d'un processeur simple, Intel(R) Xeon(R) CPUE5335. Les processeurs Intel fonctionnent à2.00GHz, avec512Kode RAM. Le système d'exploitation est Ubuntu 12.04(x86-64-linux-gnu). Afin d'assurer la fonction de filtrage, nous exécuterons des instances Netfilter au niveau des firewalls virtuels avec les mêmes règles de filtrage pour tous.

c. Résultats et discussions

i.Attaque déni de service :

La Figure 24, représente notre testbed. Nous avons déployé le logiciel Iperf sur deux machines physiques: l'une représentant le web server et l'autre le client. Ceci nous permet de prendre des mesures de la latence ainsi que du taux de pertes des paquets de notre solution de bout en bout puis de comparer les résultats.

Pour cette comparaison, nous avons tout d'abord collecté les informations des paramètres réseaux d'une architecture dite « basic[2] ». Puis nous les comparons avec les résultats obtenus avec l'utilisation de notre service avec deux principaux algorithmes de Load-Balancing qui sont :

- Round-Robin : il maintient une liste de serveurs et transmet une nouvelle connexion au serveur suivant dans la liste des membres.
- Least connexion algorithme : tient un registre des connexions serveurs actifs et transmet une nouvelle connexion au serveur avec le moins de connexions actives.

[2] Basic : architecture sans utilisation du service Cloud-based firewalling.

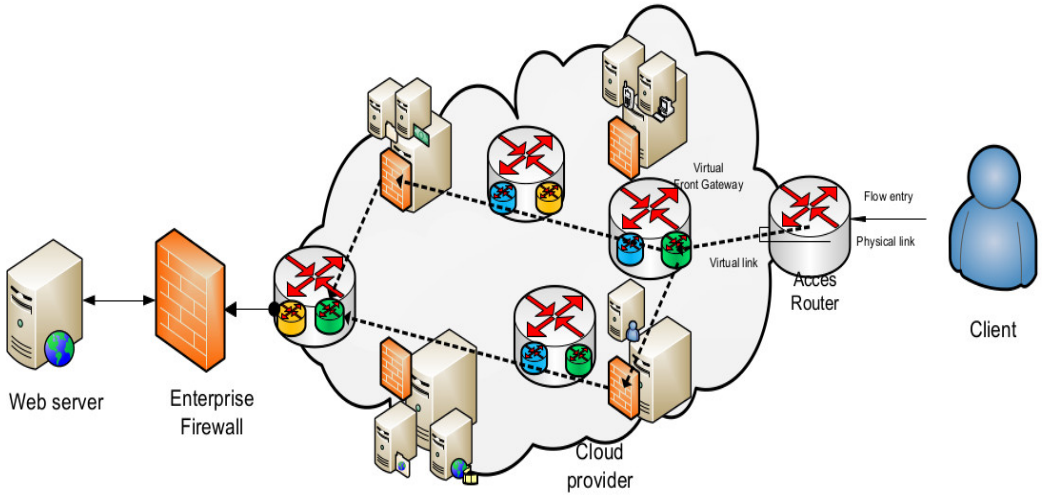

Figure 24- Architecture test-bed

La Figure 25 présente les mesures de la latence obtenues dans le cas d'une attaque DDOS périodique. L'axe des X indique le temps en secondes. L'axe des Y indique la latence. Nous analysons les résultats pour chaque intervalle de 30 secondes.

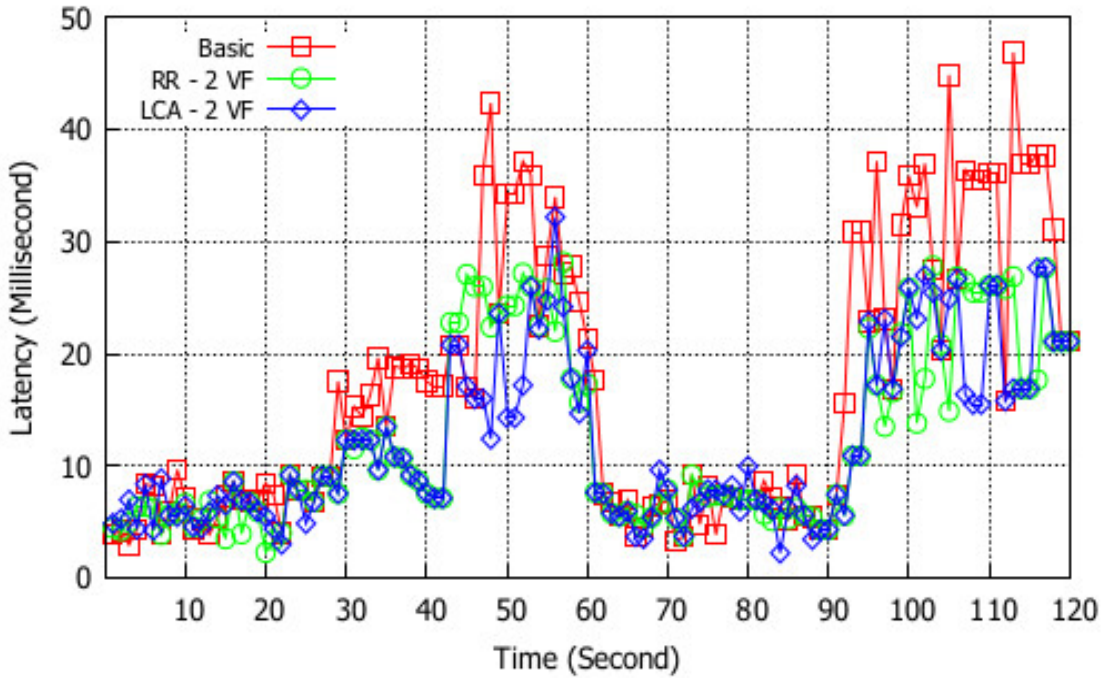

Figure 25- Latence sous attaque DDoS (deux instances)

- Absence d'attaque DDoS [0s :30s] : Dans cet intervalle, l'attaque DDos n'est pas encore exécutée. Nous observons que la latence est approximativement similaire à celle de l'architecture de référence pour les deux algorithmes.

- Présence d'attaque DDoS [30s : 60s] : Nous remarquons clairement l'augmentation des valeurs de la latence pour les différents cas avec cependant des pics importants pour l'architecture de référence. L'algorithme LCA montre de meilleurs résultats avec une diminution de presque 30% de la latence en comparaison avec l'architecture de référence. L'algorithme Round Robin vient se positionner en moyenne des deux.
- Absence d'attaque DDoS [60s : 90s] :nous pouvons voir un retour à la normale dans cet intervalle avec quasiment les mêmes chiffres que pour le premier intervalle.
- Présence d'attaque DDoS [90s : 120s] : le DDOS est appliqué à nouveau pendant 30 secondes. Nous nous apercevons que les deux algorithmes de répartitions de charge obtiennent quasiment les mêmes valeurs et font baisser la latence de plus de 10ms ce qui représente un gain d'environ 25%.

Nous constatons avec ce résultat que le service proposé améliore les performances plus précisément la latence dans le cas d'une attaque et fournit le même niveau de service qu'une architecture de base en exploitation normale avec seulement deux instances de firewalls virtuels. Donc le service apporte un plus, une valeur ajoutée. En effet, cela est confirmé par la Figure 26, qui représente le même scénario de tests mais avec 3 instances de pare-feu virtuels toujours disposés en parallèle. On remarque un gain allant jusqu'à 70%, ce qui nous permet de conclure que le nombre croissant d'instances de pare-feu utilisés a une incidence sur l'amélioration des performances.

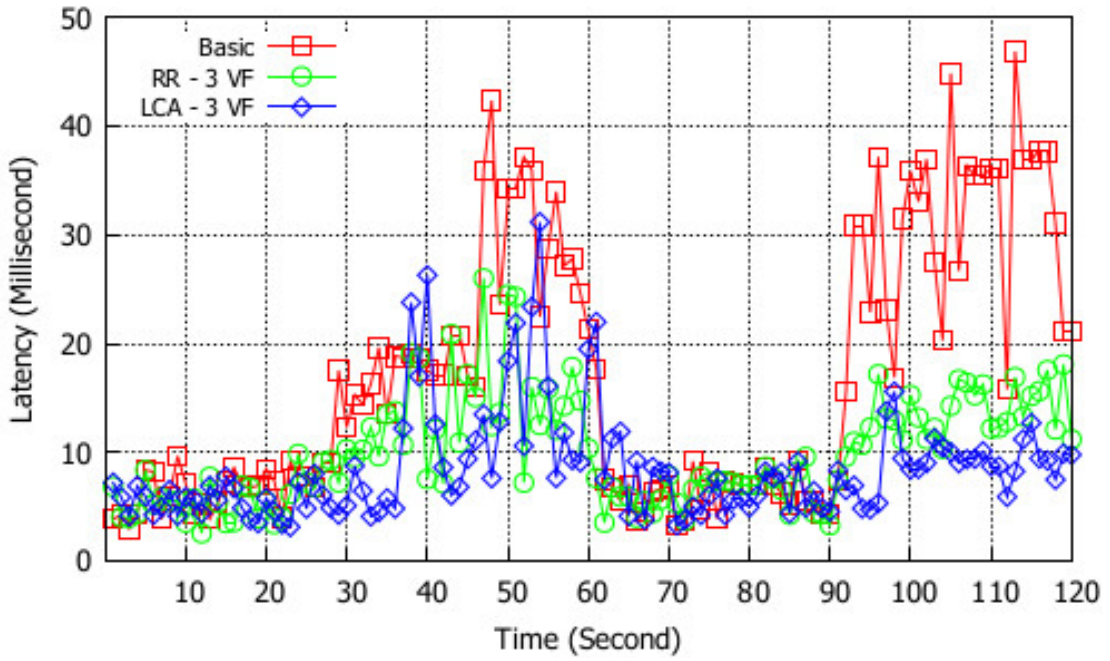

Figure 26- Latence sous attaque DDoS (trois instances)

Nous avons aussi pris en compte le taux de pertes de paquets pendant cet intervalle de temps de 120 secondes et les résultats sont présentés dans la Figure 27. On remarque bien que l'on a un taux de pertes de paquets nul lors de l'attaque DDoS, alors que l'architecture de référence est de 100% de pertes.

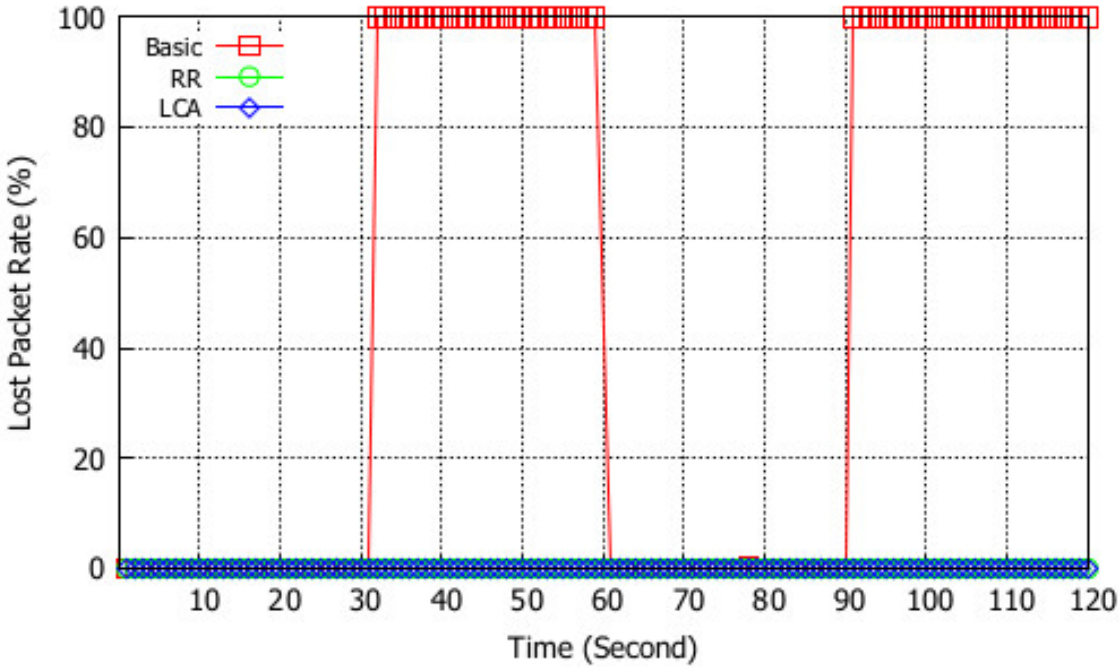

Figure 27- Pertes de paquets sous attaque DDoS

En conclusion, nous avons testé la proposition de service de Firewalling Cloud-Based sous une grande contrainte qui est l'attaque DDoS. L'architecture a montré un excellent comportement avec une grande amélioration des performances tant sur la latence que sur le taux de pertes des paquets en comparaison avec l'architecture de référence.

ii.La congestion réseau :

Nous allons démontrer comment le service de Firewalling Cloud Based permet d'obtenir une bonne fluidité de trafic en éliminant les points représentant des goulots d'étranglement dans le réseau. Pour rappel, cette solution a été proposée pour faire face aux limitations de ressources qui caractérisent les pare-feu physiques et qui représentent un goulot d'étranglement pour le réseau de l'entreprise lorsque celui-ci faisait face à une augmentation de trafic. Maintenant avec l'utilisation de ce service, le pare-feu physique ne recevra que le trafic légitime et équivalent à sa capacité de traitement.

Pour démontrer cette fluidité, nous augmentons le trafic généré pour saturer la bande passante allouée pour les besoins de l'architecture jusqu'à atteindre les 100% de saturation. On augmente à chaque fois de 10% le trafic et on enregistre le temps de parcours moyen d'un paquet dans le cas d'une architecture de référence, puis dans l'architecture du service proposé, avec à chaque fois deux instances de pare-feu puis trois. Les résultats obtenus sont discutés par la suite.

Dans la Figure 28, l'axe des Y représente la mesure de la latence du réseau (RTT: Round-trip delay) et l'axe des X représente le pourcentage de saturation de la bande passante

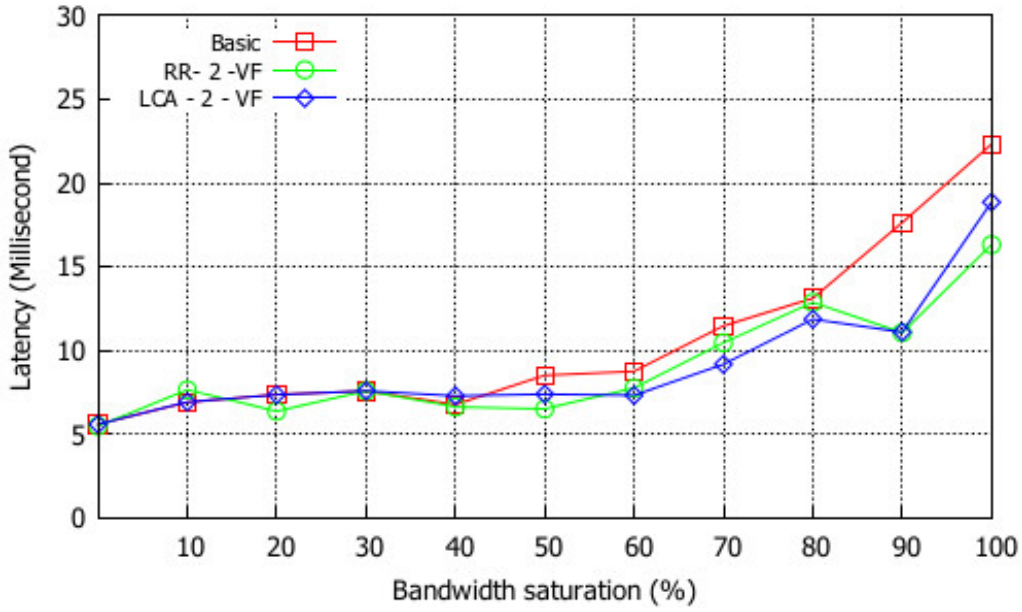

Figure 28- Latence avec saturation de bande passante (2 instances)

La Figure 28 représente les valeurs de la latence lors de la saturation de bande passante pour l'architecture de base et les deux algorithmes de Load-Balancing (Round-Robin et LCA). Les résultats montrent un comportement plus ou moins similaire avec l'architecture de référence mais avec tout de même une diminution de la latence de 5 à 10 ms pour 100% de saturation. Cette baisse est confirmée dans la Figure 29, lors de l'utilisation de 3 instances de pare-feu virtuels avec ici une baisse de 40% à saturation maximale.

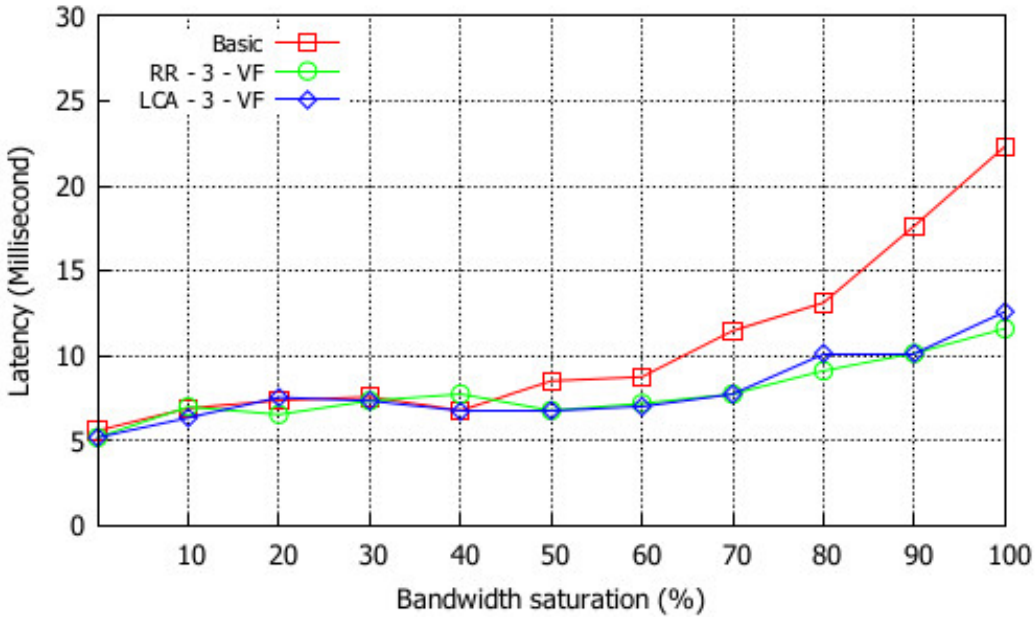

Figure 29- Latence avec saturation de la bande passante (3 instances)

Nous pouvons conclure que le service de Cloud-Based Firewalling permet surtout de proposer une **fluidité** de trafic, car la latence est sujette à la localisation géographique du couple Cloud-Provider et Client. Cette architecture permet de se prémunir des goulots d'étranglements engendrés par les limitations de ressources des équipements physiques en proposant d'épurer le trafic en amont.

5. Conclusion :

Dans ce travail, nous sommes partis de l'idée d'utiliser la grande capacité de calcul et de ressources du Cloud et les techniques de pare-feu parallèles pour concevoir un modèle de service de sécurité. Ce dernier est entièrement Cloud-Based utilisant des machines de sécurité possédant d'importantes ressources pour faire face aux attaques DDoS. Pour ce faire, nous avons déployé une architecture complète composée de 3 éléments principaux : la Front-Gateway, Les instances de pare-feu virtuels et la Back-Gateway.

Les résultats obtenus ont démontré les aptitudes de ce service à faire face à des attaques réseaux de type Flooding et à augmenter la capacité d'analyse en distribuant le trafic sur plusieurs pare-feu virtuels. Les résultats englobent la latence (RTT) et le taux de pertes des paquets dans le réseau.

Cependant lors du déploiement de cette architecture, nous avons constaté des manques comme l'automatisation des opérations de monitoring, l'allocation dynamique des ressources en instanciant et l'arrêt des instances de pare-feu au besoin. Ceux sont ces aspects que nous avons traités dans le chapitre suivant.

V. Système multi-agents pour le management des opérations du service Firewalling Cloud-Based:

Dans ce chapitre, nous présentons un système multi-agents pour la gestion et la répartition de trafic dans notre solution de pare-feu virtuels ; cette architecture virtuelle prend en charge la création et la suppression automatisées des firewalls virtuels en fonction du trafic à analyser, fournissant ainsi une gestion dynamique et à la demande de l'architecture de sécurité présentée précédemment.

Nous avons proposé dans la partie précédente, une architecture complètement virtualisée se composant d'une Front-Gateway qui représente le point d'entrée de l'architecture. Celle-ci a pour mission de prendre en charge le trafic entrant (Internet) et de le distribuer vers les pare-feu virtuels les plus adaptés (Figure 30).La Front-Gateway peut prendre la décision d'instancier ou de supprimer à la demande et/ou aux besoins un nouveau pare-feu virtuel spécialisé. Elle se base pour cela sur la densité du trafic à analyser mais aussi sur d'autres paramètres comme le type de trafic à analyser. Les paquets sont filtrés (analysés) au sein de la solution Cloud. Le trafic légitime est alors routé vers le réseau de l'entreprise. Celle-ci prendra la responsabilité de l'accepter sans l'analyser et ainsi faire totalement confiance au Cloud Provider ou non. Cette architecture est un système de défense supplémentaire à moindre coût, composé de pare-feu virtuels dynamiques localisés dans le Cloud.

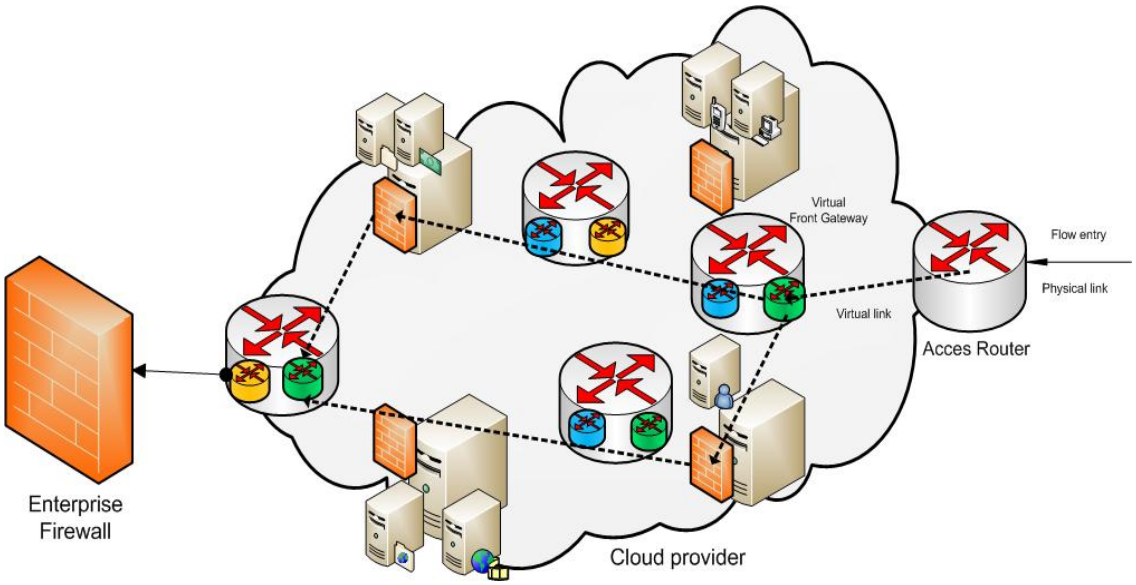

Figure 30- Cloud Based firewilling service

Notre approche est basée sur un système multi-agents qui collecte et agrège toutes les informations distribuées tant au niveau de la passerelle et que celui des firewalls virtuels. Le SMA applique un algorithme dynamique et centralisé d'équilibrage de charge pour répartir le trafic efficacement entre les différents pare-feu virtuels.

Dans un SMA de nombreux agents coopèrent pour atteindre un objectif qui dépasse leurs propres capacités respectives. Chaque agent peut être défini [56]comme:

« ...une entité logicielle qui fonctionne en permanence et de façon autonome dans un environnement particulier... en mesure de mener des activités d'une manière souple et intelligente qui est sensible aux changements de l'environnement... Serait en mesure d'apprendre de son expérience…un agent qui habite un environnement avec d'autres agents peut être en mesure de communiquer et de coopérer avec eux... ».

1. Architecture du Système Multi-agents :

L'architecture de gestion que nous présentons dans ce document est basée sur un système multi-agents. Nous avons choisi d'utiliser des agents basiques avec des objectifs relativement simples et clairs pour garantir une utilisation efficace de notre système. La Figure 31 représente les principaux agents du système. Nous pouvons diviser les agents en deux principales catégories dépendamment de leurs objectifs.

L'agent de communication et les agents externes ont comme objectif de créer la meilleure perception possible de l'environnement pour l'agent de décision. L'agent de communication met à jour les différents états de chaque agent, puis enregistre les informations au niveau du module de perception. D'autre part, l'agent externe se concentre sur la récupération et l'envoi des informations critiques.

L'agent de décision quant à lui a pour objectif de choisir la stratégie la plus optimale en se basant sur une liste de priorité préalablement mise à jour pour distribuer dynamiquement le trafic entrant vers les firewalls virtuels en équilibrant la charge de manière efficace.

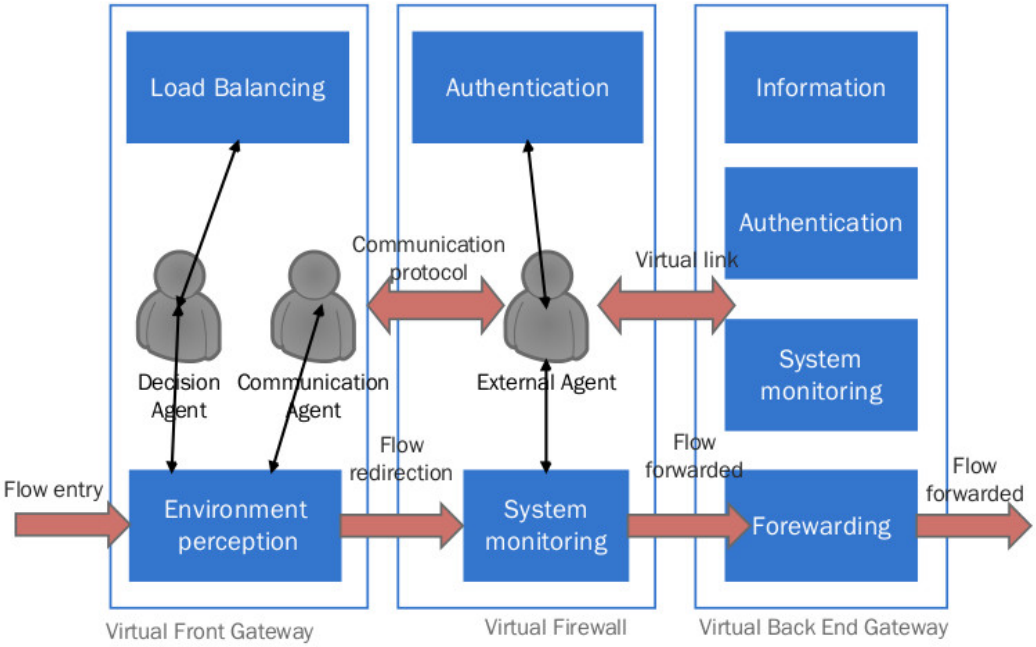

Figure 31- Framework du System Multi-Agents

a. <u>L'agent de décision:</u>

Cet agent est exécuté au niveau de la Front-Gateway. Il utilise les informations stockées au niveau du module de perception ce qui lui permet de créer et/ou mettre à jour la liste de priorité. L'agent de décision utilise la liste de priorité pour décider de la stratégie à appliquer en termes d'équilibrage de charge du trafic à traiter. La Figure 32 représente les différents composants logiciels de l'agent de décision.

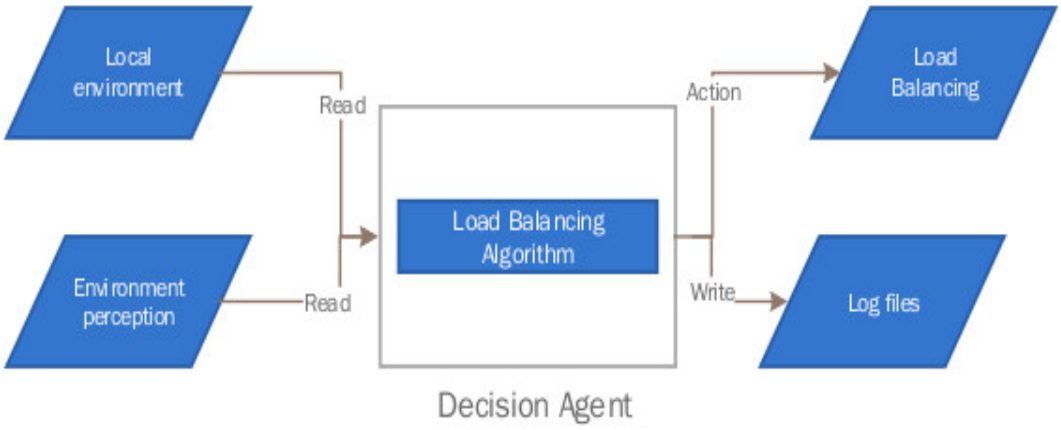

Figure 32- Composants de l'agent de décision

L'algorithme suivant (Figure 33) détermine le fonctionnement général de l'agent de décision. Il prend en compte comme entrée les paramètres locaux et globaux enregistrés dans le module de perception. L'agent doit réguler le nombre de pare-feu virtuels et la stratégie à appliquer quant à l'équilibrage de charge.

Data: Local and global parameters
Result: Load balancing between virtual firewalls
initialization;
while *true* **do**
    read CPU usage;
    read Perception Module;
    if *the entry flow > the general throughput of the virtual firewalls;* **then**
        Order the communication agent to create new virtual firewall;
    end
    Execute the session dynamic load balancing algorithm;
end

Figure 33– Algorithme de l'agent de décision

b. L'agent de communication :

L'agent de communication est l'orchestrateur du système global de communication de l'architecture proposée plus précisément, la communication entre la Front-Gateway et les différentes instanciations des pare-feu virtuels. Il enregistre les informations récoltées dans le module de perception. La Figure 34 présente les différentes actions de l'agent de communication: obtention de notifications et écriture dans le module de perception pour l'utilisation future de l'agent de décision.

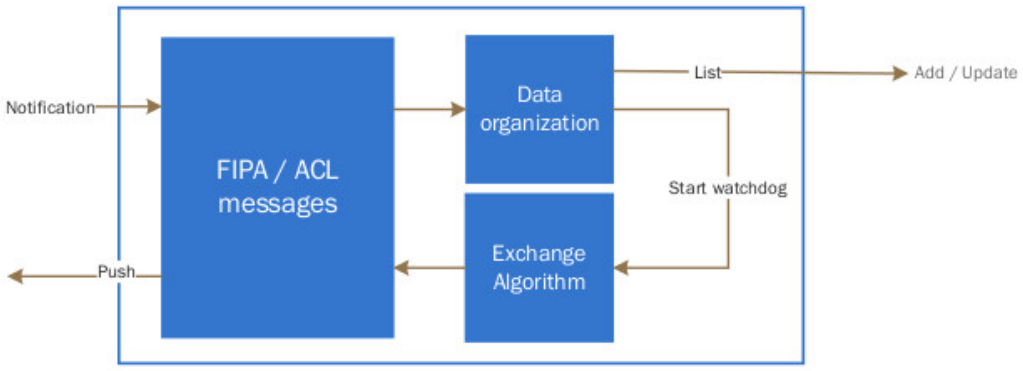

Figure 34- Composants de l'agent de communication

L'algorithme ci-dessus décrit le fonctionnement de l'agent de communication ; l'agent est en charge de la mise à jour du module de perception en réception/émission des messages de mise à jour des agents externes. Il peut aussi créer un nouvel agent externe à la demande du module de décision.

```
Data: Notifications
Result: Update the perception module
initialization;
while true do
    Connect to each virtual firewall in SSL;
    if notification received is not redundant || push not
    answered then
        Updates the Perception Module;
    end
    if any notification in delta t from the agent then
        Send a push message;
    end
    if received order to create new VM then
        create new External agent;
    end
end
```

Figure 35- Algorithme de l'agent de communication

c. Agent externe :

Chaque pare-feu virtuel doit héberger un agent externe. Cet agent, comme le pare-feu est exécuté (respectivement détruit) à l'instanciation (respectivement suppression) du firewall virtuel. Son objectif est de recueillir des métriques (CPU, Bande passante…etc.) propre au pare-feu s'y associant. Puis, de les envoyer à l'agent de communication via des notifications comme représenté dans la Figure 34.

Nous avons choisi d'utiliser un protocole basé sur des messages de type 'Push' et 'Pull' :

- Push : équivalent à une requête: Ce type de message ne peut être envoyé que par la Front-Gateway (agent des communications). L'agent suppose que les informations récoltées par un Push précédent sont toujours d'actualité pendant un certain $\Delta Temps$ assez élevé, Principalement car l'agent suppose que les machines virtuelles sont d'une très grande disponibilité, sauf dans le cas d'une réception d'un message Pull.

- Pull : ce type de message est une alarme envoyée à l'agent de communication dans le cas d'une erreur grave dans le système et/ou réseau du pare-feu virtuel hôte ne lui permettant plus d'accepter de trafic.

Le protocole garantit une bonne synchronisation entre les deux parties: la Front-Gateway et les pare-feu virtuels (par des messages Pull) et réduit les messages échangés avec les exigences minimales (par messages push).

L'algorithme de l'agent externe (Figure 36) reprend tous les procédés utilisés par le pare-feu virtuel (agent externe) :

- Monitoring des paramètres réseaux et systèmes.
- Evaluation préliminaire des informations
- Répondre au Push envoyé par l'agent de communication
- Transmettre une notification en cas d'erreur. Ceci permet de ne pas surcharger le réseau avec un trafic de synchronisation par exemple qui n'est pas utile.

```
Data: Local parameters
Result: evaluation and sending
initialization;
connection to communication agent;
while true do
    read CPU usage;
    read network state;
    evaluate data;
    send data;
    if system error then
        send notification
    end
    if push received then
        send response
    end
end
```

Figure 36- Algorithme de l'agent externe

d. Fonctionnement général du système multi-agents :

Nos différents agents qui composent notre SMA fonctionnent en collaboration pour dans un premier temps, recueillir les informations nécessaires et à l'obtention de la perception optimale de tout l'environnement. Ce qui permet à l'agent de décision

d’éditer les stratégies les plus adaptées. Les étapes nécessaires à l'exécution de cette méthode sont les suivantes:

- Instanciation de la Front-Gateway pour un trafic spécifique
- Création de l'agent de décision et de l'agent de communication
- Instanciation d’un nombre de pare-feu virtuels
- Les agents de communication recueillent les informations sur son environnement local
- L'agent de communication se connecte automatiquement aux pare-feu virtuels existants en utilisant le protocole SSL
- Chaque agent externe collecte les métriques du pare-feu virtuel correspondant et les transmet à l'agent de communication
- L'agent de communication traite puis stocke les informations dans le module de perception.
- L’agent de décision opte pour une stratégie d'après les données dans le module de perception, crée une liste de priorité correspondante à la stratégie choisie ; celle-ci (liste) permet de rediriger le flux vers les différents firewalls virtuels de façon optimale.

Nous proposons un modèle de fonctionnement qui représente notre système multi-agents qui englobe la Front-Gateway et les pare-feu virtuels. Les tâches des agents sont réparties en modules. Il existe donc trois types de tâches: Evaluation, Communication et Décision.

La Figure 37 représente la tâche de chaque agent et la coordination nécessaire pour optimiser le choix de stratégie de l’agent de décision.

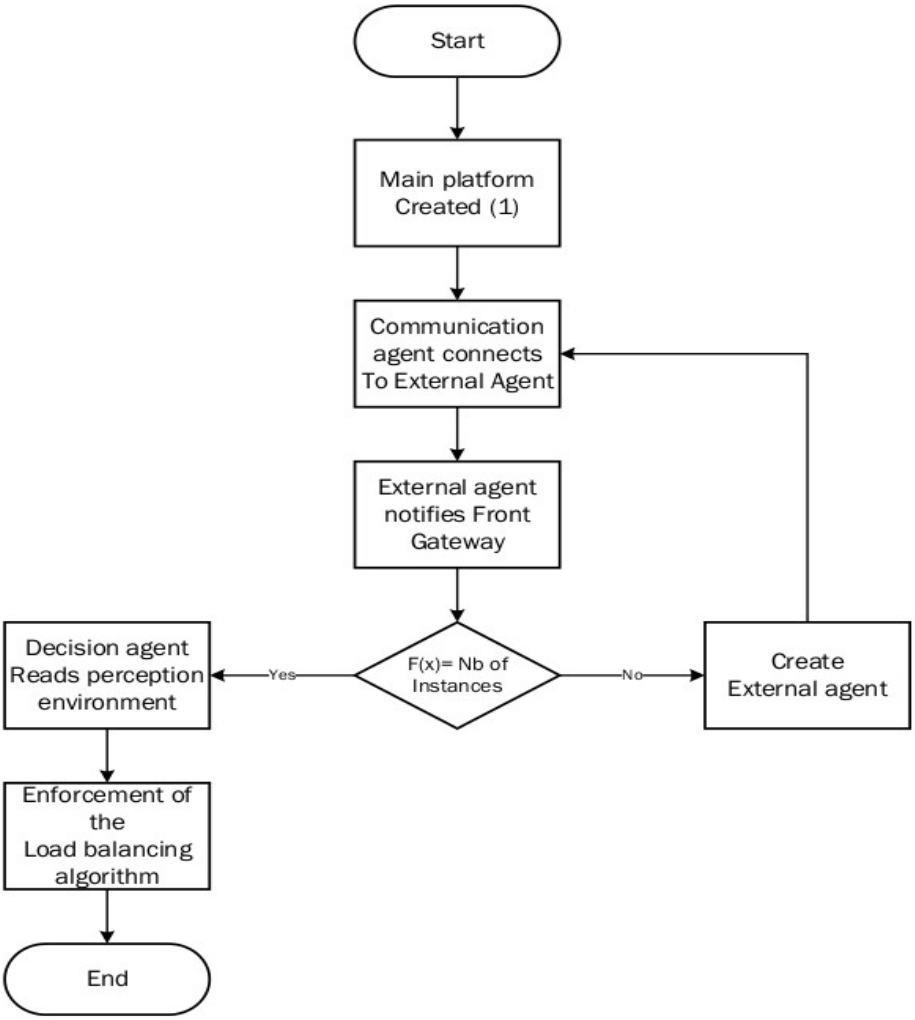

Figure 37 Flow Chart du système multi-agents

2. Implémentation du système multi-agents avec JADE:

JADE (Java Agent DEvelopment framework) [57] est une plateforme de développement d'applications multi-agents avec les spécifications FIPA (Foundation for Intelligent Physical Agents)[58]. JADE est un middleware qui implémente une plate-forme Agent ; il utilise l'environnement d'exécution Java. JADE fournit en outre:

- Un environnement d'exécution pour les agents JADE.
- Des bibliothèques de classes pour créer des agents utilisant l'héritage et la redéfinition des comportements.
- Une boîte à outils graphique pour le suivi et la gestion de la plate-forme d'agents.

Dans cette partie du chapitre, nous présentons l'architecture logicielle de notre modèle de SMA. Nous avons programmé notre modèle JADE sur le modèle UML représenté dans la Figure 38.

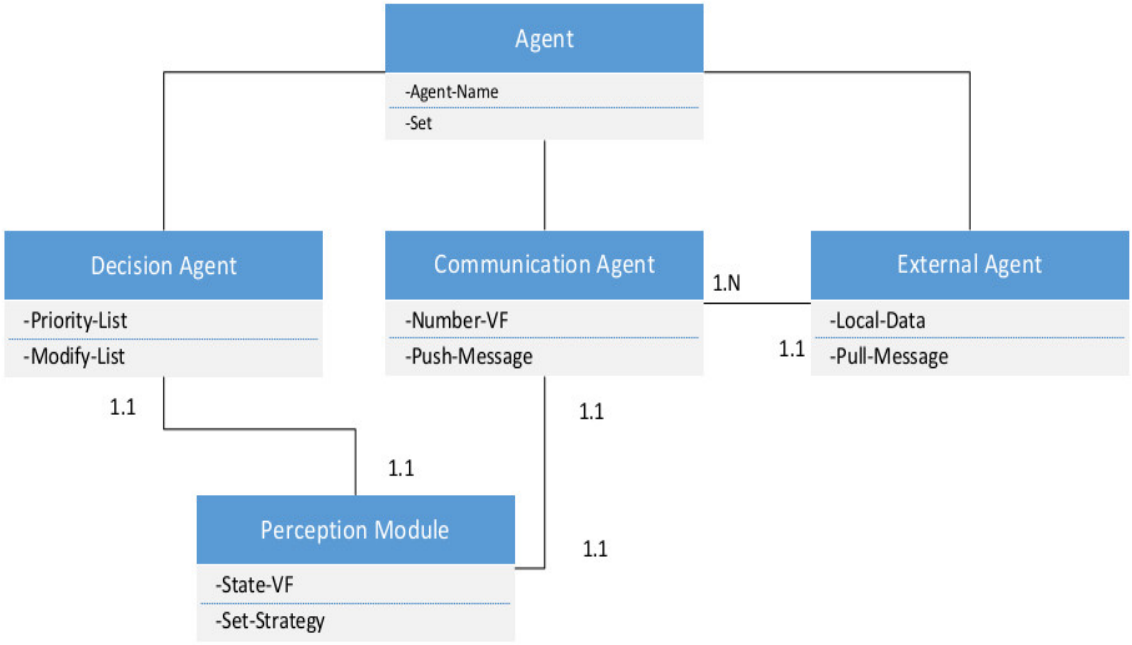

Figure 38- Système Multi-agent UML

Les principales exigences du système multi-agents de management du service Cloud-Based Firewalling et plus précisément de la fonction partage de charge (Load-Balancing)[59] sont :

- Objectif général: Un service d'équilibrage de charge doit faire peu ou pas d'hypothèses sur les types d'applications sur la charge qu'il équilibre.
- Transparence: Un service d'équilibrage de charge doit équilibrer les charges de manière transparente aux applications clientes (et aussi transparente que possible pour les serveurs).
- Adaptive: Un service d'équilibrage de charge doit être en mesure d'adapter ses décisions d'équilibrage de charge en fonction des variations de charge dynamique.
- Évolutive et extensible: Un service d'équilibrage de charge doit fournir l'évolutivité d'une application distribuée en utilisant les ressources informatiques disponibles pour gérer un grand nombre de demandes des clients et de gérer plusieurs serveurs de manière efficace et optimale.

a. Architecture logicielle de l'agent de communication :

L'agent de communication est responsable de la collecte des informations sur l'environnement global et doit mettre à jour l'état de chaque pare-feu virtuel pour ne pas encombrer la bande passante par des messages de synchronisation. L'agent de communication organise, stocke les données reçues au sein du module de perception.

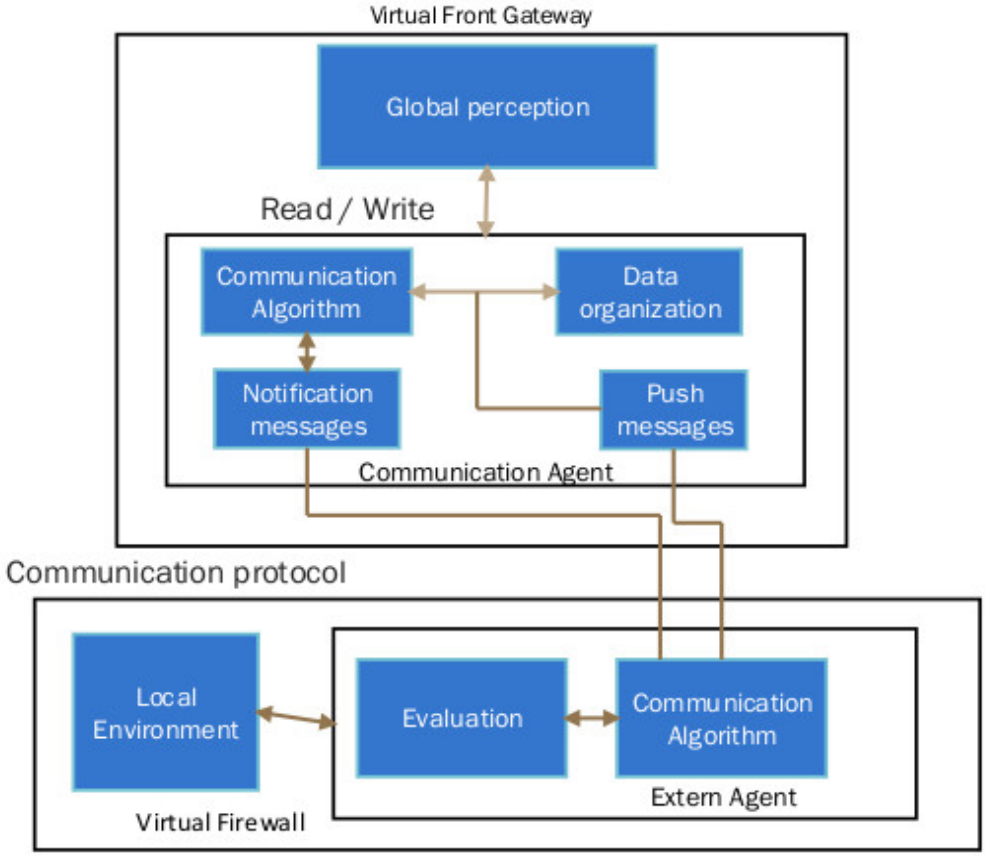

Figure 39 - Architecture logicielle de l'agent de communication et module de perception

L'agent de communication recevra en entrée des notifications des agents externes placés sur les pare-feu virtuels et/ou des messages de type Push pour mettre à jour l'état d'un agent externe n'ayant pas envoyé de notification pendant le $\Delta Temps$. L'information reçue sera traitée et organisée en forme de tableau, puis envoyée au module de perception. Un tableau correspondant principalement aux informations spécifiques que l'on observe et demande d'un agent externe. Ainsi, pour chaque tableau généré l'agent de communication réinitialise le $\Delta Temps$.

b. Architecture logicielle du module de perception :

Le module de perception représenté sur la Figure 40, est l'étape transitoire entre l'agent de décision et l'agent de communication ; il a été conçu de façon à faciliter la mise en place de nouvelles stratégies pour de futurs besoins. Ce module est composé d'un ensemble de données: la liste des stratégies et celle des paramètres à prendre en compte. Un programme gère les entrées pour accepter un ensemble de données cohérentes (Input Check Module). La liste des paramètres peut contenir des informations telles que : l'adresse IP, le taux de perte, le débit, les états et le nombre de pare-feu connectés.

Le module de contrôle d'entrée dans la Figure 40 exécute un programme qui filtre les entrées. Il choisit de mettre à jour ou non une ligne existante dans le tableau ou bien d'en ajouter une.

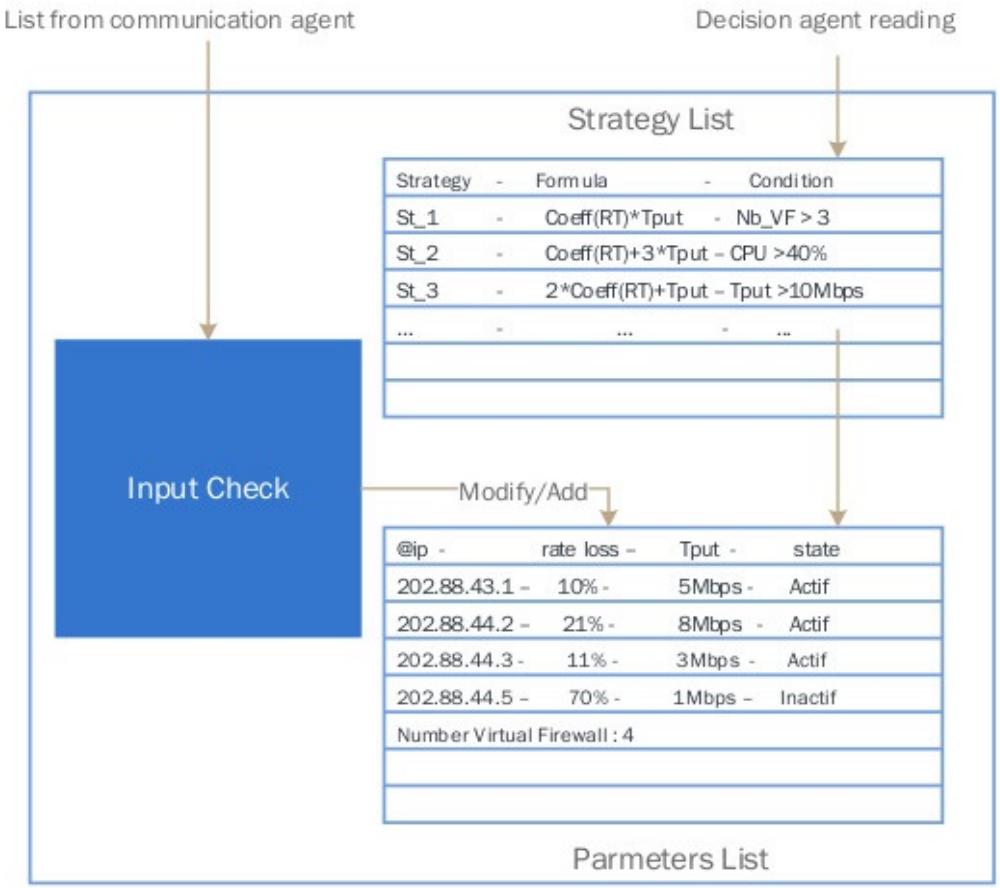

Figure 40- Architecture logicielle du module de perception

c. Architecture logicielle de l'agent de décision :

L'agent de décision représenté par Figure 41 est le principal agent de notre SMA. Il dispose de deux types d'informations en entrée :

- Informations locales : l'information que l'agent peut lire sur son environnement
- Informations globales : données et stratégies organisées dans le module de perception.

Il crée et/ou met à jour la liste de priorité qui lui permet d'adapter ou modifier la stratégie à appliquer en terme d'équilibrage de charge du trafic à analyser.

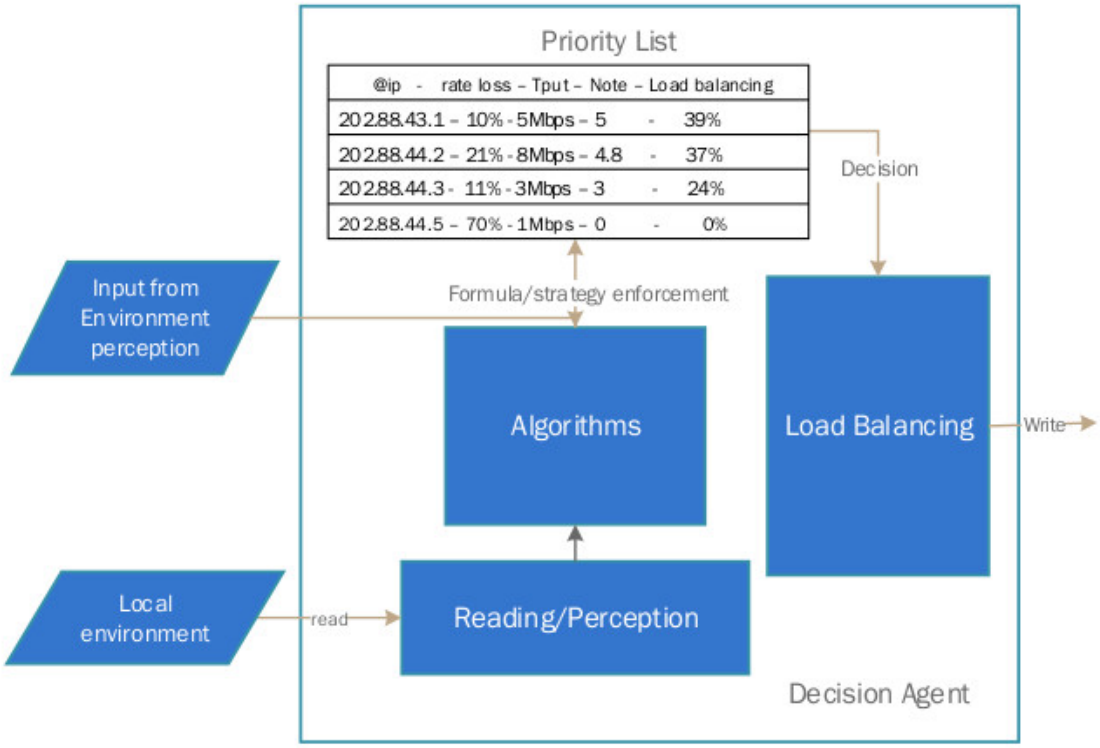

Figure 41- Architecture logicielle de l'agent de décision

i.Algorithme de Load-Balancing :

L'algorithme d'équilibrage de charge est exécuté par l'agent de Décision ; il s'agit d'un algorithme centralisé et dynamique, qui repose sur l'utilisation des états d'information pour améliorer la décision. C'est une entité centrale (basée dans l'agent de décision) et qui prend en charge la décision pour le système global.

L'agent de décision fonde sa décision sur les informations locales et globales du module de perception. Dans notre modèle, la librairie de stratégie peut être facilement augmentée et adaptée pour différents besoins et services. Cela permet d'assurer une bonne adaptation et interopérabilité avec des technologies futures.

Dans notre travail, nous avons utilisé trois critères pour évaluer l'état d'un pare-feu virtuel dans un contexte Cloud: le taux de perte des paquets, le Throughput et le nombre de règles de filtrages (RF). Nous utilisons deux formules précises : l'une pour noter la Front Gateway et l'autre pour noter les liens (chemin) entre la Front Gateway et les instances de pare-feu virtuels qu'on nomme respectivement $fact(lien)$ et $fact(Front\_Gateway)$ normalisées entre 0 et 1.

- Définition de $fact(lien)$:

$$Link_i = \frac{Throughput_i}{Taux\ de\ pertes_i}$$

$$fact(Link_i) = \begin{cases} \dfrac{Link_i}{Link_i + 1} & si \quad Taux\ de\ pertes_i \neq 0 \\ \dfrac{Throughput_i}{Throughput_i} + 1 & sinon \end{cases}$$

- Définition de $fact(Front\ Gateway)$:

$$\text{Front\_Gateway} = \frac{Throughput_{FG}}{Contraintes_{FG}}$$

$$sachant\ que\ Contraintes_i = Nombre\ de\ RF$$

$$fact(\text{Front\_Gateway}) = \frac{\text{Front\_Gateway}}{\text{Front\_Gateway} + 1}$$

Au niveau de l'agent de décision, nous avons la possibilité de mettre à jour ou de modifier les formules utilisées et de les mettre en adéquation avec le service à protéger. Cependant, lors de nos tests nous n'utiliserons que la formule ci-dessus.

ii. Mécanisme d'identification du nombre d'instances de firewalls virtuels :

Notre système doit décider automatiquement de créer une nouvelle instance de pare-feu virtuel ou de supprimer celles déjà existantes dépendamment du trafic à traiter. Cela se fait en fonction de nombreux paramètres ; ces derniers dépendent soit de l'état

du système de la Front-Gateway et ses performances et/ou de la capacité du lien. En effet, le Throughput que l'on obtient à la sortie de la Front-gateway est diminué comparé à l'input, et les liens peuvent subir des pertes et l'on exprime avec le coefficient β .On exprime ces deux caractéristiques via une fonction nommée $fact$ (que nous avons défini plus haut).

Nous définissons la fonction f qui nous donne en résultat le nombre d'instances de pare-feu comme suit :

$$f = Nb_{VM} = \frac{T_{source}}{T_{vm}} * \beta \; sachant \; que \; \beta \in [0,1]$$

$T_{source} =$ est le débit à traiter par le service. exprimée en bits par seconde (bps)

$T_{vm} =$ est le débit maximum que traite chaque instance.

Nous supposons que toutes les instances ont la même capacité de traitement. Tous les liens entre la Front-Gateway et les instances ont le même débit. Donc, Nous définissons β en fonction de $fact$ comme suit :

$$\beta = fact\ (link) * fact(node)$$

3. Conclusion :

Notre modèle de système multi-agents est une automatisation simple et efficace de la distribution du trafic entre les pare-feu virtuels qui composent notre architecture générale. Le choix d'un modèle multi-agents améliore la réactivité et l'adaptabilité rapide aux fluctuations et aux changements dans l'environnement Cloud. Notre système est adaptatif et laisse la possibilité de mettre en place des optimisations futures, que cela soit au niveau des agents, du protocole de communication ou bien des algorithmes utilisés.

La Figure 42 reprend notre modèle SMA, il décrit l'objectif et la fonctionnalité principale de chaque agent.

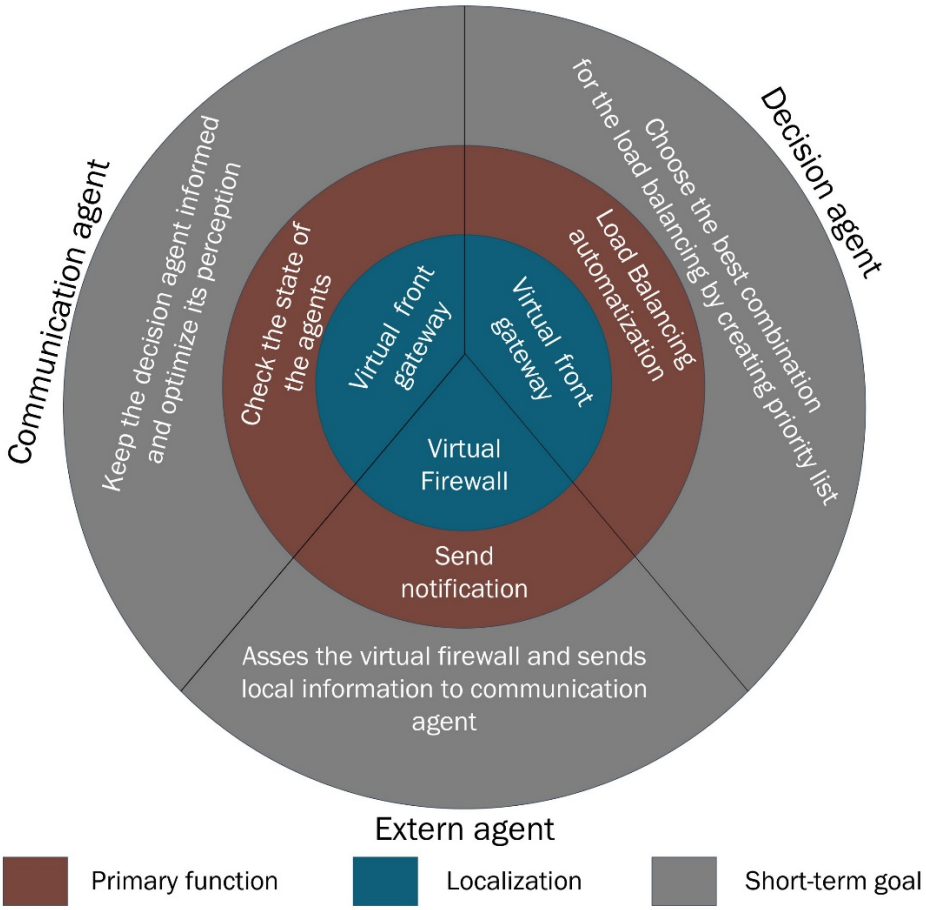

Figure 42- Représentation générale du Système Multi Agent

VI. Conclusion générale :

1. Contributions :

Le Cloud Computing a évolué au cours de la dernière décennie, passant d'un simple service de stockage à des services plus complexes, en proposant le software comme service (SaaS), les plateformes comme service(PaaS) et très récemment la sécurité comme service (SECaaS).Dans notre travail, nous sommes partis de l'idée simple d'utiliser les ressources offertes par le Cloud avec un faible coût financier pour proposer des nouvelles architectures de service de sécurité.

En premier lieu, nous avons proposé une architecture hybride pour offrir du Cloud-Based Firewalling service. Cette architecture a eu pour but d'augmenter la puissance de calcul des pare-feu physiques. Elle est proposée avec deux schémas de déploiement « Secure Forwarding Architecture » et « Secure Sharing Architecture » que nous avons testés dans deux environnements réels qui sont: l'attaque DDoS et la congestion du réseau. Après comparaison des résultats, nous concluons que le service proposé peut faire face à des attaques distribuées de déni de service et peut être déployé comme un service anti-DDoS.

Après avoir identifié les lacunes de l'architecture hybride, nous avons pris l'initiative de proposer une architecture complètement Cloud-Based (présentée dans le chapitre 4). Elle permet à un Cloud provider de proposer un service de Firewalling à ses clients. Celui-ci leur demande de s'abonner à l'offre en leur garantissant le traitement (analyse) d'une capacité de bande-passante de trafic avec des règles de filtrages fonctionnelles et d'autres proposées par l'abonné. Par la suite, le Cloud Provider joue le rôle d'intermédiaire pour le client et ne lui transmet que le trafic légitime, ce qui permet à l'utilisateur d'avoir une capacité de traitement supplémentaire sans avoir à acheter, déployer et maintenir de nouveaux équipements.

Pour concevoir un modèle de service de sécurité, nous avons couplé des techniques de Load-Balancing avec les techniques de pare-feu parallèles. Ce qui nous a permis, d'obtenir d'excellents résultats qui ont démontré que notre service fait face à des attaques réseaux de type Flooding, et d'augmenter la capacité d'analyse en distribuant le trafic sur plusieurs pare-feu virtuels. Cependant, la gestion d'un tel service a besoin de répondre aux contraintes de manière autonome. Dans ce but, nous avons

proposé un système multi-agents (chapitre 5) qui est une automatisation simple et efficace de la distribution du trafic entre les pare-feu virtuels qui composent notre architecture générale. Le choix d'un modèle multi-agents améliore la réactivité et l'adaptabilité rapide aux fluctuations et aux changements dans l'environnement Cloud.

2. Perspectives :

Ce travail ouvre la voie à plusieurs perspectives de recherche.

A court terme, nous prévoyons d'améliorer notre architecture multi-agents pour une meilleure automatisation de la répartition du trafic entre les pare-feu qui prendrait en compte le type de trafic et d'autres paramètres de qualité de service. Nous allons également améliorer la stratégie d'allocation des ressources "à la demande" des pare-feu virtuels. Pour répondre cette question, de nouveaux mécanismes d'élasticité doivent être conçus pour gérer l'approvisionnement et la réallocation des ressources de manière autonome.

Un autre point de discussion c'est la distribution des politiques de sécurité et des règles de filtrages entre les différentes instances de pare-feu virtuels qui peuvent parfois prêter à confusion. En effet, des mécanismes de distribution existent mais sont-ils adaptés aux environnements Cloud ? Doit-on modifier ces mécanismes en conséquence ou bien en proposer de nouveaux. Il faut aussi prendre en compte la modélisation du trafic dans ce type d'architecture ce qui permettra d'adapter la réaction de l'architecture aux comportements des différents types de trafic à traiter.

Un nouveau type d'attaques est en train de voir le jour nommé Cloud-Internal Denial of Service ; c'est un déni de service interne spécifique au Cloud où un groupe d'utilisateurs (attaquants) coordonnent leurs charges de travail pour attaquer certaines ressources du Cloud et déclencher la migration des services, et donc obtenir un temps d'inaccessibilité du service. Nous pensons que l'on peut adapter nos architectures Afin de prendre en compte ce type de menaces et arriver même à les contrer.

Liste des publications :

Conférences internationale avec acte et comité de lecture :

1. *Guenane, F.A; Samet, N.; Pujolle, G.; Urien, P., "A strong authentication for virtual networks using EAP-TLS smart cards," Global Information Infrastructure and Networking Symposium (GIIS), 2012 , vol., no., pp.1,6, 17-19 Dec. 2012.*
2. *Guenane, Fouad Amine; Pujolle, Guy, "Strong virtual network authentication using EAP-TLS smart-cards," Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on Cloud Networking, vol., no., pp.197,199, 28-30 Nov. 2012.*
3. *Msahli, M.; Pujolle, G.; Serhrouchni, A; Fadlallah, A; Guenane, F., "Openflow and on demand networks," Network of the Future (NOF), 2012 Third International Conference on the Network of the Future , vol., no., pp.1,5, 21-23 Nov. 2012.*
4. *Guenane, F.; Dumas, P.Y.; Nogueira, M.; Pujolle, G., "Solving virtual network resource allocation problems using a constraint satisfaction problem model," Network of the Future (NOF), 2013 Fourth International Conference on the Network of the Future , vol., no., pp.1,5, 23-25 Oct. 2013.*
5. *Guenane, Fouad; Boujezza, Hajer; Nogueira, Michele; Pujolle, Guy, "An architecture to manage performance and reliability on hybrid cloud-based firewalling," Network Operations and Management Symposium (NOMS), 2014 IEEE , vol., no., pp.1,5, 5-9 May 2014.*

Papiers en cours de soumission :

1. *Multi-Agent System for Load Balancing in Cloud Based Firewalling Service.*
   Guenane, F. Bendriss, J. Nogueira, M. Pujolle, G.
   The International Conference on Network and Service Management (CNSM).
2. *Reducing DDoS Attacks impact using a Hybrid Cloud-Based Firewalling Architecture. Guenane, F. Nogueira, M. Pujolle, G.*
   The Global Information Infrastructure and Networking Symposium (GIIS).
3. *Autonomous Architecture For Managing Cloud-Based Service*
   Guenane, F. Bendriss, J. Nogueira, M. Pujolle, G.
   The Network of the Future conference.
4. *DDOS Mitigation Cloud-Based Service.*
   Guenane, F. Bendriss, J. Nogueira, M. Pujolle, G.
   IEEE Globecom 2014 Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA).
5. *Novel Hybrid Architecture to manage Cloud based Firewalling services.*
   Guenane, F. Ayadi, I. Nogueira, M. Pujolle, G.
   Journal of Network and Systems Management.

Liste des illustrations :

Annexe

Annexe-1 : Allocation de ressources physiques dans les réseaux virtuels :

La virtualisation des réseaux doit assurer une sensation de liberté de mouvements à tous les niveaux. Chaque réseau virtuel doit être libre de mettre en œuvre sa propre topologie,ses fonctionnalités de routage, ainsi que ses protocoles de contrôle personnalisés indépendamment du réseau physique sous-jacent et d'autres réseaux virtuels coexistants. Plusieurs axes de recherches existent pour le domaine de la virtualisation des réseaux, interfaçages et l'ergonomie d'utilisation pour les administrateurs des réseaux virtuels, la signalisation et l'amorçage des systèmes. Il est vrai que lors de la création d'un nouveau réseau virtuel ou l'ajout d'un nœud dans un réseau existant a un effet sur la converge du réseau qui englobe le temps d'instanciation du nœud ou du réseau, additionné à cela le temps de mise à jour des tables de routage, la découverte des ressources et des topologies.

le fournisseur d'infrastructure physique doit être capable de déterminer et de posséder l'information la plus actuelle possible de l'état de son infrastructure et des ressources physiques consommées par les réseaux instanciés. Le contrôle d'admission, le monitoring, l'exploitation, la sécurité, la confidentialité, interopérabilité, l'adressage sont autant d'aspects en cours de traitement.

Fondamentalement, plusieurs travaux dans le domaine ont été entrepris par différentes équipes [60][61][62][63][64]. Cependant, l'allocation des ressources physiques reste un problème ouvert dans le domaine de la virtualisation des réseaux [65], plus précisément celui de l'optimisation de l'allocation des ressources offertes par le réseau physique et celui de l'attribution de la bande passante sans violation de ces contraintes. Il s'agit de trouver le bon équilibre entre les besoins du client, les contraintes liées à l'état du réseau (en terme de Latence, architecture, Qualité de service, etc.).

Dès lors, même si diverses contraintes et architectures (topologie) rendent ce problème de calcul très complexe, il laisse intacte la problématique et nous permet ainsi de proposer des algorithmes et des solutions plus optimales. Plusieurs travaux dans le domaine n'ont pas pris en compte toutes les propriétés de l'allocation (présentées plus loin) ; certains se focalisent sur la connaissance de requêtes virtuelles à l'avance, d'autres ont été réalisés d'une manière centralisée.

Notre travail de recherche a pour but d'élaborer une solution pour l'allocation dynamique des ressources en la modélisant comme un problème de satisfaction de contraintes. Nous présentons le concept de virtualisation en présentant la virtualisation système avec les différents mécanismes, en poursuivant vers la virtualisation réseau ses avantages et inconvénients ainsi que les performances que l'on peut attendre de cette technologie. Un état de l'art sur les différents axes de recherches dans le domaine de la virtualisation, en se focalisant principalement sur la problématique de l'allocation des ressources qui représente l'objectif principal de notre projet. Nous présentons par la suite l'implémentation de notre modèle et solution avec la librairie Choco de java ; cette librairie offre une ou plusieurs classes permettant de programmer des problèmes de satisfaction de contraintes suivis par les résultats obtenus et une conclusion.

1. Etat de l'art de l'allocation des ressources dans les réseaux virtuels :

Dans un environnement de virtualisation de réseau, un certain nombre de réseaux virtuels coexistent sur le même réseau physique. Chaque réseau virtuel est donc composé d'un sous-ensemble des ressources du réseau physique sous-jacent.

L'allocation des ressources est effectuée par les fournisseurs d'infrastructure (InP) sur réception d'une requête de la part d'un VN. Cette phase est également connue comme VN **mapping**. Le scénario le plus simple est celui où [61] considère la topologie VN (nœuds virtuels et liens virtuels), la capacité de calcul (CPU), la mémoire et l'emplacement des nœuds virtuels est dependent de la bande passante des liens virtuels. Dans ce qui suit, nous présentons une formulation mathématique du problème d'allocation des ressources, puis nous passons en revue les travaux antérieurs sur l'allocation des ressources en matière de virtualisation réseau.

a. Problème d'Allocation de ressources :

La création d'un réseau virtuel nécessite l'allocation des ressources physiques pour ses nœuds et ses liens en respectant un certain nombre de propriétés. Si nous représentons un réseau virtuel et son réseau physique correspondant par deux graphes, on peut considérer que le problème d'allocation est un problème de Mapping des ressources physiques au réseau virtuel.

Plusieurs algorithmes de Mapping ont été déjà proposés afin d'allouer des ressources physiques d'une façon efficace. Nous les présenterons plus loin dans ce

document, l'article [60] est pour nous le liminaire du domaine ; il définit les bases d'une instanciation d'un réseau virtuel, mais il propose aussi un modèle du problème d'allocation des ressources.

Ainsi, plusieurs travaux postérieurs s'inspirent de ce modèle et des résultats qui s'y affèrent. En effet, [60] propose des règles à suivre pour l'instanciation d'un réseau virtuel. Or, le processus de création des réseaux virtuels commence après avoir effectué la virtualisation des ressources physiques. Par conséquent, un groupe de ressources virtuelles qui est créé et représenté par une couche virtuelle qui met en œuvre l'abstraction des ressources physiques disponibles. Ce réseau virtuel sera sujet à trois étapes liées :

- Description des ressources.
- Découverte des ressources.
- Provisionnement des ressources

Toutes les requêtes des réseaux virtuels sont prises en charge et tout le processus peut être contrôlé et administré ; la figure dans l'article représente les étapes associées au processus d'instanciation.

Pour rappel, les buts conduits par la politique de management du réseau virtuel sont :

- Le système doit permettre aux utilisateurs de réserver des ressources à travers le réseau pour des opérations prévisibles et fiables.
- Le système doit assurer une isolation suffisante, de manière à éviter les interférences entre utilisateurs.
- Le système doit avoir des mécanismes de contrôle d'admission de sorte que seul le nombre limité de demandes qui sont en file d'attente peuvent recevoir des services et donc éviter la congestion.

Dès lors, nous avons étudié le modèle mathématique pour mieux comprendre les tenants et aboutissants et ainsi voir si les objectifs peuvent être atteints avec ce dernier.

[60]représente la topologie physique par un graphe $G_v(i) = \{V^v(i), E^v(i)\}$, on peut décomposer tout le processus en deux problèmes :

- Assignement des nœuds $f_n(i) : \{V^v(i), C_n^v\} \rightarrow \{V', R_n\}$
- Assignement des liens $f_l(i) : \{E^v(i), C_l^v\} \rightarrow \{E', R_l\}$

Où$V' \subset V^s$, $E' \subset E^s$ avec R_n et R_l représentent les ressources allouées à l'ieme demande du réseau virtuel.

Il faut savoir que les problèmes d'assignement des nœuds et des liens sont dépendants, il faut les traiter simultanément.

b. Objectifs des fonctions d'instanciation des réseaux virtuels :

L'objectif principal est clair ; le processus d'instanciation du réseau virtuel doit être économique en matière de ressources physiques. L'allocation des ressources doit s'effectuer avec l'idée d'optimisation de ces dernières (ressources).

Une notion de limite de nœuds et de liens a été introduite pour l'assignement d'un nombre maximal de ressources attribuées. Aucun processus pour le choix des limites n'a été prouvé à ce jour.

La maximisation des gains (services) est générée par l'instanciation du réseau virtuel en adéquation avec la politique commerciale du Provider. Les fonctions doivent prendre aussi en compte la surcharge du CPU et de la bande passante des nœuds utilisés pour ne pas créer une surcharge du trafic déjà existant. Ces objectifs ne prennent pas en compte la flexibilité des liens et leur fiabilité

c. Instanciation des nœuds est un NP-problème :

Assignement des nœuds du réseau physique pour le réseau virtuel sans la violation des contraintes de bande passante est un NP-Problème (défini dans l'introduction de [60]). Pour résoudre ce souci trois approches ont été identifiées : force-brute backtracking algorithm, simulated annealing et algorithme d'approximation. Le premier n'est pas applicable à grande échelle (nombre important de nœuds). Pour conclure cette partie, le problème est très complexe, une solution efficace et évolutive pour instanciation des nœuds doit être trouvée.

Ce modèle nous a semblé très intéressant car il prend en considération beaucoup d'aspects de la problématique, et on peut incorporer d'autres propriétés et d'autres éléments comme des systèmes multi-agents que l'on verra plus tard pour solutionner le problème de gestions des ressources.

Par la suite, pour simplifier les problématiques de l'allocation des ressources, plusieurs travaux n'ont pas pris en compte toutes les propriétés indiquées plus haut,

certains se focalisent sur la connaissance de requêtes virtuelles à l'avance comme [61][62].

Les axes fondamentaux du problème d'instanciation des réseaux virtuels sont:

- l'optimisation de l'allocation des ressources offertes par le réseau physique en respectant ses contraintes
- le respect du cahier des charges imposé par le "Service Level Agreement" (SLA), notamment la qualité de service (QoS)

Une complexité est ajoutée lorsque l'aspect environnemental est pris en compte. En effet, la possibilité d'économiser de l'énergie peut être intégrée à l'algorithme en tant que contrainte supplémentaire. Quel que soit le contexte, toute allocation proposée doit être capable de prendre en charge le trafic souhaité (exigé). Si nous représentons un réseau virtuel et le réseau physique par deux graphes, on peut considérer le problème d'allocation comme un problème de matching du réseau virtuel aux ressources physiques. L'objectif principal est clair, le processus d'instanciation du réseau virtuel doit être économique en matière de ressources physiques. L'allocation des ressources doit s'effectuer avec l'idée d'optimisation de ces dernières.

Lors de notre recherche bibliographique, nous avons identifié plusieurs algorithmes que nous catégorisons comme suit :

- Approche centralisée: Une entité centrale est responsable pour mapper les réseaux virtuels au réseau physique. Elle doit maintenir les mises à jour des informations sur le réseau physique (les ressources disponibles) afin de prendre les décisions appropriées pour allouer des ressources. Cette approche pourrait souffrir de problèmes d'évolutivité et de passage à l'échelle. De plus, la communication entre l'entité centrale et les autres nœuds du réseau physique (mise à jour des informations sur les ressources disponibles) peut faire subir une surcharge réseau considérable. Plus la taille du réseau augmente plus graves sont les problèmes mentionnés ci-dessus.
- Approche distribuée : pour faire face aux problèmes de l'approche centralisée, le processus d'allocation des ressources peut être réparti sur tout ou partie des nœuds physiques dans l'InP. En règle générale, chaque nœud physique impliqué dans l'allocation des ressources utilise ses connaissances locales à cet effet. Les

protocoles de communication et la coopération sont nécessaires pour coordonner le processus.

En plus des approches centralisées, et décentralisées, il existe deux autres approches, il s'agit de :

- Approche statique (sans reconfiguration): ne permet aucun changement dans l'assignement des ressources durant le temps de vie du réseau virtuel. L'algorithme d'allocation des ressources est celui de l'assignement sans reconfiguration.
- Approche dynamique (avec reconfiguration): adaptative, elle permet les changements d'allocation des ressources en dépendance avec les demandes et les performances du réseau virtuel. . L'algorithme d'allocation des ressources est celui de l'assignement avec reconfiguration.

2. Problématique :

Les axes fondamentaux du problème d'instanciation des réseaux virtuels sont l'optimisation de l'allocation des ressources offertes par le réseau physique en respectant ses contraintes ainsi que le respect du cahier des charges imposé par le "Service Level Agreement" (SLA), notamment la qualité de service (QoS)

Une complexité est ajoutée lorsque l'aspect environnemental est pris en compte. En effet, la possibilité d'économiser de l'énergie peut être intégrée à l'algorithme en tant que contrainte supplémentaire. Quel que soit le contexte, toute allocation proposée doit être capable de prendre en charge le trafic souhaité (exige). Si nous représentons un réseau virtuel et le réseau physique par deux graphes, on peut considérer le problème d'allocation comme un problème de matching du réseau virtuel aux ressources physiques. L'objectif principal est clair, le processus d'instanciation du réseau virtuel doit être économique en matière de ressources physiques. L'allocation des ressources doit s'effectuer avec l'idée d'optimisation de ces dernières.

Les méthodes de résolution d'un problème de matching de graphe peuvent être divisées en deux grandes classes [66]:

- Les méthodes **exactes** et les méthodes **approchées**.
    - La méthode exacte suppose qu'il existe un sous-graphe et que sa tâche est de trouver ce dernier mais dans certaines

situations où les données sont altérées, une correspondance parfaite peut ne pas être trouvée.

- La méthode approchée, à l'instar de l'autre cherche une correspondance optimisant la fonction objective du matching. Dans notre étude les différents algorithmes qui s'inspirent de ces méthodes (Ullmann, Nauty, Schmidt et Druffel) ne correspondent pas parfaitement à notre problème. En effet, le matching de réseaux doit être exact pour les capacités des nœuds et liens.

Plusieurs définitions découlent de cette problématique, ce qui nous permet de la conceptualiser avec un modèle; ce dernier nous permettra de clarifier les hypothèses, et ainsi pouvoir la solutionner plus aisément.

On nomme un graphe pondéré un quadruplet G = (V, E, φ_V, φ_E) où V est l'ensemble des sommets, E ⊂ V×V est l'ensemble des arêtes, φ_V : V⟶ N^p est la fonction de pondération des sommets et φ_E : E⟶ N^q est la fonction de pondération des arêtes. P et q sont les nombres de caractéristiques significatives respectivement pour les sommets et les arêtes.

Il est à noter que différentes contraintes découlent de cette définition et qui rendent le modèle choisi plus cohérent pour notre problème ; un nœud virtuel ne peut être associé qu'à un seul nœud physique; l'inverse n'est pas juste, car un nœud physique et donc un routeur physique peut contenir plusieurs machines virtuelles. Il en est de même pour les liens (arêtes) qui eux aussi sont sous le coup de contraintes, un lien physique existe entre deux machines seulement et il ne peut par contre contenir plusieurs liens virtuels. De là, il en découle qu'un lien virtuel ne peut relier que deux nœuds virtuels.

Il est aussi à mettre en évidence qu'un lien virtuel est aussi un chemin qui peut traverser plusieurs nœuds physiques pour mettre en liaison les nœuds virtuels alloués sur ces derniers (nœuds physiques).

En conclusion plusieurs liens physiques peuvent être associés à un lien virtuel. Ceci représente les contraintes absolues de notre modèle.

a. Matching de graphe :

Représentation	Signification
N^S	Ensemble des nœuds du réseau physique
L^S	Ensemble des Liens du réseau physique
A_N^s	Propriétés du nœud physique N
A_L^s	Propriétés du lien Physique L
N_v	Ensemble des nœuds du réseau virtuel
L_v	Ensemble des Liens du réseau virtuel

Tableau 2- Notations utilisées dans le modèle

Nous modélisons le réseau physique par un graphe pondéré et non orienté tel que $G^s = (N^s, L^s, A_N^s, A_L^S)$où $N^s\ et\ L^s$ représentent respectivement l'ensemble des noeuds et liens du réseau physique, ces noeuds possédent chacun des propriétés A_N^s que nous avons identifiés comme la capacité CPU ainsi que la mémoire, il en va de même pour les liens A_L^S dont la propriété principale est la capacité en termes de bandes passantes. Nous prenons aussi en compte les chemins existants et le fonctionnement entre les nœuds du réseau physique et les représentons avec une matrice que l'on nomme P^s.

L'instanciation du réseau virtuel se distingue par une requête qui se modélise aussi par un graphe pondéré non orienté, $G^V = (N^V, L^V, C_N^V, C_L^V)$on peut constater que le couple N^V, L^V représente la topologie logique du réseau virtuel. Le réseau virtuel se caractérise par des contraintes sur les nœuds et liens virtuels qui le constituent, C_N^V, C_L^V modélisent respectivement les contraintes liées au nœud N et le lien L.

L'allocation des ressources physiques au réseau virtuel n'est rien d'autre que le matching de G^V sur une partie de G^S, tout en respectant les contraintes du réseau virtuel à instancier :

$$f: G^V \rightarrow (N', P', R_N, R_L)$$

La fonction d'allocation globale représente une application qui a pour domaine de départ G^V , et domaine d'arrivée qui est constitué de:

- $N' \subset N^S$: l'ensemble de noeuds alloués,
- $P' \subset P^S$: l'ensemble des chemins, un chemin peut être composé de plusieurs liens, l'inverse n'est pas juste physique entre deux nœuds virtuels alloués.
- R_N, R_L : respectivement les ressources des nœuds et liens alloués.

Il faut savoir que les problèmes d'assignement des nœuds et des liens sont dépendants, il faut les traiter simultanément.

- Assignement des nœuds:

$$f^N: (N^V, C_N^v) \rightarrow (N', R_N)$$

- Assignement des liens:

$$f^L: (L^V, C_l^v) \rightarrow (P', R_L)$$

L'analyse des différents articles traitant des performances des outils de virtualisation nous a menés à mettre en place deux paramètres supplémentaires qui sont S_n etS_l respectivement la Charge du nœud N et du Lien L, utilisés dans [63] et dont le but est de quantifier les ressources consommées dans l'instanciation du réseau virtuel. Nous les avons repris dans le but d'améliorer les performances. En effet, il est préférable qu'il ne soit pas confié un nombre important de liens virtuels à un lien physique qui serait perturbé (coupure, saturation...etc.), de même qu'un routeur sur lequel on aurait instancié un nombre élevé de machines virtuelles se verrait diminué, tout ceci reste des expériences métiers qui nous orientent, dans l'usage de certains paramètres d'utilités que cela soit pour les nœuds et/ou les liens dans le but:

- D'améliorer les performances du réseau physique par répercussion celles des réseaux virtuels qui sont mappés dessus,
- Faire baisser les taux de pertes des paquets, en faisant baisser la collision des paquets dûe à la circulation croisée de plusieurs flux dans les liens physiques

Il est nécessaire d'avoir un partage de charge sur tout le réseau substrat (physique), ce qui comprend les nœuds et les liens. Ceci est l'un des moyens mis en œuvre pour satisfaire le SLA.

b. Utilité client, Utilité Provider

Le Service Level agreement (SLA) est un contrat entre un fournisseur de services, dans notre cas un fournisseur de réseau physique est un client (réseau virtuel) [67]. Ce contrat spécifie quels services doit fournir le provider au client et les pénalités si ces derniers ne sont pas respectés.

Les attentes des clients dans la plupart des études de marché soulignent l'exigence suivante [67][68]:

- Mesure fiable de la qualité des services (Qos)
- Fourniture de la qualité attendue des services.
- Optimisation de l'utilisation des ressources.

Dans le domaine de la virtualisation un SLA peut être produit entre différents protagonistes, il existe dans le domaine des télécoms à ce jour, différents types d'intervenants:

- Carrier Service Provider : opérateur d'infrastructure physique qui a pour mission principale l'interconnexion des différents réseaux opérateurs.
- Internet Service Provider (ISP): on peut aussi dire Fournisseur d'Accès Internet (FAI), il permet à des individus ou à des entreprises de se connecter à Internet.
- Opérateur: est une entité qui met à disposition de ses clients des services de communication à distance (mobile, internet,...etc.).
- Entreprise/utilisateur: Consommateur final des services qui lui sont proposés par l'ASP ou l'ISP.

Le Tableau 3 montre clairement les interactions possibles entre ces acteurs pour la mise en place d'un service de virtualisation. Un Carrier Service Provider qui possède une infrastructure physique très importante peut proposer à un ISP ou à un opérateur son infrastructure physique pour instancier des réseaux virtuels à la demande ; l'ISP ou l'opérateur quant à lui peut alors proposer ce service (virtualisation des réseaux) à ses clients, le principe est celui de Client/Fournisseur.

Client	Fournisseur
Carrier Service Provider	Carrier Service Provider
Internet Service Provider	Carrier Service Provider
Entreprise/ opérateur	Internet Service Provider
Utilisateur	Internet Service Provider

Tableau 3- Acteurs du Service Level Agreement

L'application opérationnelle du SLA au niveau des technologies réseaux se traduit par l'adéquation des différents paramètres qui le constituent aux besoins du client ; ces paramètres sont présentés dans le Tableau 4.

P	Paramètres
P1	Pertes de paquets
P2	délai
P3	gigue
P4	bande-passante
P5	CPU, Mémoire
P6	Topologie
P7	Sécurité/Disponibilité
P8	controle d'admission

Tableau 4- Liste des paramètres SLA

Tous les paramètres ne sont pas nécessaires pour décrire les besoins d'un utilisateur ; si nous prenons pour exemple un fournisseur de service VOIP sera intéressé par les paramètres P_1, P_2, P_3 utilisés pour définir la Qos dont il a besoin, à l'inverse un site de e-commerce s'intéressera à P_4, P_7 des paramètres plus globaux et qui correspondent mieux à ses besoins.

Nous prenons en compte dans notre modélisation les paramètres $P_{i\ telque\ i \in [1,7]}$ car nous nous intéressons à 3 caractéristiques, et nous ajoutons à cela une notion de temps de vie:

• Qualité de service: nous prenons en compte

- Le délai d'acheminement d'un paquet en mode End-to-End: il est convenu que le temps d'acheminement du réseau physique doit être inférieur ou égal à celui demandé dans le SLA, plus ou moins un certain Δ_t (préalablement convenu)

$$G(N,L): T_{Rep} = T(n_i, n_j) | \ \forall\ n_i, n_j \in N \times N: T_{rep} \leq\ Delai_{SLA} \pm \Delta_t$$

- Le taux de perte des paquets dépendra de la qualité des lignes empruntées et du dimensionnement du réseau, on le notera$q_{substrat}$, on obtient :

$$q_{substrat} \leq\ q_{SLA} \pm \Delta_q$$

- La gigue est la variation du délai de transmission. l'une des causes de variation de la gigue est que les paquets n'empruntent pas

forcement le même chemin, une autre cause de la variation du délai de transit dépend du nombre de routeurs traversés et de la charge de chaque routeur traversé, on notera $g_{substrat}$:

- $g_{substrat} \leq g_{SLA} \pm \Delta_g$

Dans la majeure partie des cas, la gigue doit rester inférieure à 100 ms pour garder une qualité acceptable.

- La disponibilité du service: la disponibilité est aujourd'hui un enjeu important des infrastructures informatiques (réseau, système d'informations...etc.), elle désigne le fait que cette architecture ou ce service a un taux de disponibilité convenable. Pour mesurer la disponibilité, on utilise souvent un pourcentage qui représente le temps où le service doit être disponible. Pour le SLA différents métriques peuvent être mis en place; exemple: disponibilité par année, par mois ou par semaine. Nous notons:
  - H : un réseau physique tombe en panne,
  - $P(H)$ la probabilité que l'évènement H se réalise, sachant qu'un réseau physique est constitué d'un ensemble de routeurs physiques indépendants alors :

$$P(H) = \sum_{h \in H} P(h)$$

  - La probabilité qu'un réseau virtuel reste disponible est égale à celle de la disponibilité de tous les routeurs physiques sur lesquels il est instancié $P(X)$ est:

$$P(X) = 1 - P(H)/h \in N'$$

- Les ressources: cette partie a déjà été traitée plus haut, avec C_N^v, C_l^v qui représentent respectivement les besoins en ressources pour les noeuds et les liens convenus dans le SLA.

- Temps de vie: chaque réseau virtuel que l'on instancie possède un temps de vie, ce temps de vie est défini par:
  - $T_{debut} = instant\ d'instanciation\ du\ réseau\ virtuel$

- $T_{fin} = instant\ d'arrêt\ du\ réseau\ virtuel$

Nous avons identifié deux types de propriétés, fonctionnelles et non fonctionnelles. En effet, certaines propriétés comme la topologie et les ressources allouées sont fonctionnelles représentant des besoins et des contraintes qui doivent être rigoureusement satisfaits, inversement les propriétés non fonctionnelles correspondent à celle concernant la Qos ainsi que la disponibilité. Le tableau suivant résume les différents types.

Propriétés	P1	P2	P3	P4	P5	P6	P7	P8
Type	NF	NF	NF	F	F	F	NF	F

Non Fonctionnelle	NF
Fonctionnelle	F

Tableau 5 : Classement des propriétés par type

Un contrat doit prendre en compte deux aspects, le bien-être client et fournisseur, une fonction d'utilité doit être mise en place, pour cela nous utilisons une classification des services et sa correspondance avec le SLA, nous avons adapté la correspondance proposée par [67] à nos besoins, le tableau suivant représente cela:

	Taux de pertes	délai	gigue	Bande passante	CPU mémoire	Topologie	Disponibilité	contrôle d'admission
Voice	=	++	++	+	=	=	+	++
Vidéophone	=	++	+	++	=	=	+	++
Téléphonie	=	=	0	0	=	=	+	++
Multimédia	+	++	0	++	=	=	+	++
VOD	=	++	0	++	=	+	+	++
VPN	Dépend du types de flux encapsulés				=	++	++	++
Données temps-réel	++	++	0	+	=	=	+	++
Données (web, mail , e-Commerce)	++	+	0	0	=	=	+	++
Streaming	++	=	0	0	=	=	+	++

++	Très grande performance	=	Performance par default
+	Grande performance	0	indifférent

Tableau 6 : Mapping SLA et Service

Nous identifions donc des classes de services, les besoins d'un client correspondront à une ou plusieurs classes, notre solution calcule l'allocation la plus optimale en se basant sur la fonction objective qui lui sera définie

Le fournisseur devra mettre en place les contraintes correspondantes à tous les paramètres du SLA. Donc l'allocation des ressources doit répondre à ces conditions. De là nous décomposons le problème en deux sous problèmes, l'allocation des nœuds et l'allocation des liens.

i.Utilité Client :

Le client exprime ses besoins dans le SLA ; ses besoins dans notre modèle SLA sont exprimés par plusieurs propriétés définies pour plusieurs services bien distincts ; la principale mission est de choisir une allocation qui répondra aux besoins du client au mieux de ces propriétés, dès lors comment différencier une allocation d'une autre ? Comment savoir laquelle est la plus appropriée ? Nous décidons de baser nos prises de décisions sur les techniques de **multicritères**.

Les méthodes d'analyse multicritère ou plus exactement les méthodes d'aide multicritère à la décision, sont des techniques assez récentes et en plein développement [69] L'aide à la décision multicritère se présente comme une alternative aux méthodes d'optimisation classiques basées sur la définition d'une fonction unique. L'intérêt des méthodes multicritères est de considérer un ensemble de critères de différentes natures (exprimés en unités différentes), sans nécessairement les transformer. Par leur manière d'intégrer tout type de critères, ces procédures semblent mieux permettre de se diriger vers un judicieux compromis plutôt qu'un optimum souvent désuet.

Il s'agit en effet d'identifier et de mesurer les conséquences des actions sur lesquelles va porter la décision. L'évaluation de l'action sera donc effectuée sur un ensemble de critères. Comment alors définir le(s) critère(s) principal(aux) dans des contextes différents.

- Définition relation de préférence [69]:

La fonction d'utilité u définie dans A tel que:

$$u: A \rightarrow R^{+}$$

On suppose qu'il existe un $\varepsilon \in A$. Il existe une “zone d’insensibilité” lors de la comparaison des valeurs d'utilités u(x) et u(y). Alors la relation de préférence P s'écrira de la manière suivante:

$$x, y \in A^2 : xPy \Rightarrow u(x) - u(y) > \varepsilon$$

De même nous définissons la relation d'indifférence I, qui est la négation de la relation précédente de préférence on aura :

$$xIy = \neg (xPy)\ et\ \neg(yPx)$$

Ceci s'exprime aussi comme :

$$x, y \in A^2 : xIy \Rightarrow |u(x) - u(y)| \leq \varepsilon$$

Exemple: pour mieux comprendre la notion d'ordre d'intervalle, on prend la fonction d'utilité qui pour deux valeurs u (x_1)=125ms et u (y_1) =126ms qui expriment le délai d'acheminement d'un paquet (paramètre temps de propagations),va nous donner comme résultat une indifférence ; ce que l'on veut démontrer c'est qu'il n’y a pas de préférence entre ces deux valeurs, c'est pour cela que chaque paramètre a sa zone d'indifférence.

La définition des critères nécessite par la suite une évaluation de la contribution et l’influence de chaque paramètre dans la décision finale, dans notre cas, on doit classer (évaluer) les allocations que l’on propose, avec un ensemble de critères. Ces critères sont les paramètres non fonctionnels que nous avons définis plus haut, l'importance des propriétés dans le choix diffère par service proposé.

On prend comme exemple le paramètre délai entre les services VOIP et Streaming ; il est clair qu'il est critique pour le service VOIP, alors qu'il l'est moins pour le Streaming considéré comme un critère de moyenne importance. D'après [24,25] nous avons pu ordonner les paramètres non fonctionnels d'après leur importance et par service, ce qui a donné comme résultat le tableau suivant:

	Taux de pertes	Délai	Gigue	Disponibilité
VOIP	3	1	2	4
Vidéophone	3	1	2	4
Téléphonie	3	2	4	1

Multimédia	2	1	4	3
VOD	3	1	4	2
VPN	3	1	4	2
Données temps réels	2	1	4	3
Données	1	2	4	3
Streaming	1	3	4	2

Tableau 7 : Classement des critères par service

Si le débit et le taux de perte concernent toutes les applications, le délai et la gigue affectent plus particulièrement les applications à temps réel ou requérant une grande synchronisation. Par exemple une gigue trop élevée va affecter la synchronisation de l'annulation d'échos pour un service conversationnel

La solution pour évaluer un contrat doit s'exprimer par un vecteur $X^{\rightarrow}$ de longueur finie et dont les élements appartiennent à des domaines finis et totalement ordonnés, donc:

$X^{\rightarrow} = (x_1, x_2, \dots, x_k)_{k\in[1,4]}$ Où x_k représente un paramétre non fonctionnel correspondant à l'ordre établi par le tableau et par les services réalisés plus haut.

Ce vecteur est la méthode d'évaluation que nous décidons d'appliquer aux contrats. Pour le choix des différents $\varepsilon_{k\in[1,4]}$, il doit être laissé aux administrateurs réseaux et systèmes de l'opérateur ;la cause en est l'hétérogénéité des équipements et logiciels exploités par l'opérateur et par le client.

Le modèle que l'on vient de mettre en place va nous permettre de comparer entre deux allocations proposées ; il prend en compte les aspects fonctionnels qui sont exprimés dans le SLA, ainsi que les non fonctionnels (qui sont liés à la QOS)

Soit $x^{\rightarrow}, y^{\rightarrow}$ les vecteurs representant respectivement les allocations x, y. on dit que x est préféré à y :

$$xPy => \sum_{k=1}^{4}\left(\left(\prod_{i=1}^{k-1}(x_i I y_i)\right) \times (x_k P y_k)\right)$$

ii.Utilité fournisseur :

L'intérêt du fournisseur de réseau physique est le gain monétaire qu'il peut obtenir de l'instanciation d'un réseau virtuel. Nous identifions deux types de gains, l'un relatif au SLA et aux accords commerciaux entre lui et le client, l'autre à l'aspect économie d'énergie qui induit une économie budgétaire.

- Gain commercial:

Nous avons discuté précédemment et modélisé les relations commerciales entre Client/fournisseur ; il met en évidence les gains engendrés par la satisfaction des besoins clients par rapport au SLA, l'allocation de ressources que l'on qualifie de non fonctionnelle répondant aux besoins clients en terme de qualité de service déjà exprimée dans le SLA engendre un gain ; ce gain est soumis à la capacité de l'opérateur à fournir cette qualité et pendant un certain laps de temps. Pour cette raison nous proposons une fonction g qui calcule le gain monétaire du fournisseur et qui aura pour paramètres les ressources allouées ainsi que le temps de vie du réseau

$$g_{service(i)} = (Paramètres\ non\ fontionnels\ ; Temps\ de\ vie) = gain\ anticipé$$

- Gain d'énergie:

L'informatique génère 2 % des émissions de CO2 liées à l'activité humaine, autant que l'ensemble de la flotte aérienne mondiale (source Gartner Grouphttp://www.gartner.com/technology/home.jsp), En France, le ministère de l'environnement estime que la consommation des TIC est comprise entre 55 et 60 TWh par an, soit 13,5 % de la consommation d'électricité totale. Le green-IT, selon la définition du Journal officiel de la République Française du 12 juillet 2009, est l'ensemble des techniques de l'information et de la communication dont la conception ou l'emploi permettent de réduire les effets négatifs de l'activité humaine sur l'environnement. Additionné à cela, un contexte de volatilité des prix de l'énergie, un fournisseur doit prendre en compte les coûts liés à sa consommation, donc l'écologie et le bénéfice peuvent atteindre un but commun.

Aujourd'hui sur chaque entité informatique (hardware) qu'il soit moyen de télécommunication ou non, les constructeurs donnent en même temps que les capacités propres au matériel sa consommation énergétique. Dès lors il devient facile de connaitre

la consommation énergétique d'un routeur au Watt près. La maitrise de la consommation électrique se simplifie car on peut la quantifier.

Notre idée est de pouvoir chiffrer l'utilité fournisseur avec précision en chiffrant le gain commercial et en déduire le coup énergétique des matériels en fonctionnement.

Pour cela, on intègre une fonction $Conso_N$ qui aura pour paramètres la consommation électrique du routeur physique, le temps d'allocation de ce dernier ainsi que le prix unitaire du K-watt d'énergie. La solution prend en compte le résultat de $Conso_N$ cette dernière retourne le coût de consommation énergétique (monétaire) de son propre nœud N tel que N ∈ N'; Il suffit de chiffrer la consommation exacte d'un routeur virtuel N^v qui sera une partie de la consommation du nœud physique N sur lequel il est alloué.

$$Conso^i = c \times Conso_N$$

Sachant que

$$c = \frac{Ressources\ consommées\ par\ le\ retour\ virtuel\ i}{Ressources\ totales\ du\ routeur\ physique\ N}$$

Par suite, la fonction $Conso^v$ calcule la consommation globale de l'assignement d'un réseau i ce qui est donne:

$$Conso^v = \sum_{i \in N^v} Conso^i$$

En conclusion, l'utilité fournisseur représente le bénéfice que lui apporte l'instanciation d'un réseau virtuel, ceci peut être exprimé par:

$$U_{Fournisseur}(T_{début}, T_{fin}) = g_{service(i)} - Conso^v$$

c. <u>Conclusion :</u>

La fonction d'évaluation globale est déterminante pour définir un ordre ou ce que l'on qualifie comme classement aux allocations : cette fonction doit pouvoir déterminer efficacement la valeur d'un ensemble de configurations que peut contenir une allocation ce qui implique que si une allocation au moins est proposée lors d'un cycle du protocole, alors tout le réseau physique s'accordera pour appliquer la ou les meilleures allocations si elles sont compatibles.

3. Principe de résolution :

On qualifie généralement de « combinatoires » les problèmes dont la résolution se heurte à une explosion du nombre de combinaisons à explorer. Cette notion de problème combinatoire est formellement caractérisée par la théorie de la complexité qui propose une classification des problèmes en fonction de la complexité de leur résolution. Dans notre cas le problème de matching de graphe est considéré comme un problème combinatoire. Le problème de la recherche du plus grand sous-graphe (partiel) commun est un problème NP-difficile. [70] déclare qu'un même problème peut généralement être modélisé de différentes façons, et les problèmes de matching de graphes peuvent être formulés sous la forme de problèmes de satisfaction de contraintes.

a. Problème de satisfaction de contraintes (CSP) :

De nombreux problèmes peuvent être exprimés en terme de contraintes comme les problèmes d'emploi de temps, de gestion d'agenda, de gestion de trafic ainsi que certains problèmes de planification et d'optimisation comme le problème de routage de réseaux de télécommunication.

Un problème de satisfaction de contraintes est composé de variables, de domaines associés à ces variables et des contraintes entre des sous-ensembles de variables [71]. Le domaine d'une variable est un ensemble de valeurs qui peuvent être affectées à cette variable. Une contrainte sur un ensemble de variables est une restriction sur les valeurs qu'elles peuvent prendre simultanément (valeurs compatibles); cette restriction peut être implicite ou explicite.

i. Définition: [Mackworth, 1977] :

Un problème de satisfaction de contraintes est défini par un triplet (X, D, C) d'ensembles:

- un ensemble fini de *n* variables $X = \{x_1, x_2, \dots, x_n\}$
- un ensemble de *n* domaines finis associés à ces variables $D = \{D_1, D_2, \dots, D_n\}$
- un ensemble de m contraintes $C = \{C_1, C_2, \dots, C_m\}$est défini sur un sous ensemble de variables de X

De façon générale, résoudre un CSP consiste à affecter des valeurs aux variables de telle sorte que toutes les contraintes soient satisfaites. De façon plus précise :

- On appelle affectation un ensemble de (x_i, v_i) tels que x_i variables du CSP et v_i est une valeur appartenant au domaine $D(x_i)$ de cette variable,
- Une affectation est dite totale si elle instancie toutes les variables du problème; elle est dite partielle si elle n'en instancie qu'une partie,
- Une affectation A viole une contrainte C_j si toutes les variables de C_j sont instanciées dans A, et si la relation définie par C_j n'est pas vérifiée pour les valeurs des variables de C_j définies dans A
- Une affectation (totale ou partielle) est consistante si elle ne viole aucune contrainte, et inconsistante si elle viole une ou plusieurs contraintes,
- Une solution est une affectation totale consistante, c'est-à-dire une évaluation de toutes les variables du problème qui ne viole aucune contrainte.

b. Représentation mathématique d'une allocation :

Soit $G_s = (V_s, E_s, \varphi_{V_s}, \varphi_{E_s})$ le graphe pondéré d'un réseau substrat et $\left\{G_{V_i} = \left(V_{V_i}, E_{V_i}, \varphi_{V_{V_i}}, \varphi_{E_{V_i}}\right) \mid 0 \leq i \leq n\right\}$ un ensemble de graphes pondérés de réseaux virtualisés. On nomme **allocation** le couple formé d'une application :

A: $\bigcup_{0\leq k\leq n} V_{v_k} \rightarrow V_s$ ⇔L'union des ensembles des nœuds de chaque réseau virtuel dans l'ensemble des nœuds du réseau substrat, d'une relation binaire entre l'ensemble des arêtes du réseau substrat et l'union des ensembles des arêtes de chaque réseau virtuel:

$$P \subset \left(E_s \times \bigcup\nolimits_{0\leq k\leq n} E_{v_k}\right)$$

c. Formalisation de notre solution CSP

i.Paramètres fonctionnels :

Notre problème de satisfaction de contraintes est défini comme un triplet (X, D, C) avec pour ensemble de variables $\bigcup_{0\leq k\leq n} V_{v_k}$ à valeurs dans le domaine $D = V_s$

Les contraintes d'un tel CSP s'expriment à la fois sur les sommets et les arêtes des graphes, suivant leurs fonctions de pondération. L'ensemble C des contraintes est défini comme suit:

- Pour chaque nœud du réseau substrat, la somme des vecteurs de pondération de tous les nœuds de réseaux virtuels associés doit être inférieure selon Pareto à son vecteur de pondération propre
- L'ensemble des contraintes sur les sommets C_v est défini comme suit:

$$C_v = \bigcup_{V_s \in D} \left(\sum_{V_v \in A^{-1}(V_s)} \varphi_{V_{V_i}}(V_v) \leq_{Pareto} \varphi_{V_s}(V_s) \right)$$

Avec G_{V_i} le graphe du réseau virtuel contenant le lien v_v et $\varphi_{V_{V_i}}$ la fonction de pondération des arêtes de ce graphe.

- Pour chaque arête (v_i, v_j) d'un réseau virtuel, l'ensemble des arêtes du réseau substrat associées doit constituer un chemin de $A(v_i)$ vers $A(v_j)$:

$$C_{E_v} = \bigcup_{(v_0, v_1) \in \cup E_v} \left(\{ e_s | e_s P(v_i, v_j) \} \; est\ un\ chemin\ de\ A(v_i) \text{ vers } A(v_j) \right)$$

Avec $(a, b) \leq_{Pareto} (c, d) \Leftrightarrow a \leq c \; et \; b \leq d.$

Ainsi l'ensemble C des contraintes du CSP est l'union des précédents ensembles:

$$C = C_v \cup C_{E_s} \cup C_{E_V}$$

ii.<u>Qualité de service (Qos) :</u>

La qualité de service peut se définir comme étant « l'ensemble des phénomènes pouvant influencer les performances du service qui déterminent le degré de satisfaction de l'utilisateur de ce service ». La qualité de service sera exprimée dans un langage non technique et le plus humainement compréhensible pour l'utilisateur. Par exemple, il peut être fait usage de critères de classification basés sur des niveaux (ex. Gold, Silver, Bronze, un nombre d'étoiles, ...). A l'inverse, les paramètres utilisés entre fournisseur de service et opérateur de réseau et entre opérateurs de réseaux eux-mêmes se doivent de répondre à des exigences techniques. Les paramètres de QoS associés à un flux de données sont principalement :

- Le débit (bandwidth) du flux qui désigne la quantité d'informations écoulée par unité de temps exprimée en bit/s. Il faudra également déterminer le point de mesure auquel correspond le débit. En effet, le débit indiqué n'aura pas la même valeur suivant la couche (au sens modèle ISO) où il est mesuré, du fait des diverses encapsulations. Le

débit est également associé à la notion de bande passante. Dans la suite du mémoire, la bande passante sera utilisée pour désigner la capacité d'écoulement d'un lien ou d'un équipement, et le débit pour exprimer le volume de données émises ou nécessaires par une application ou un service par unité de temps.

- Le taux de perte (loss rate et error rate) qui désigne la probabilité maximale de perte de données ou de paquets. Ce paramètre, sans unité, est bien entendu très inférieur à 1. On cherchera toujours à se rapprocher d'un taux de perte égale à 0 qui désigne une QoS excellente.
- Le délai (delay) de transmission, exprimé en ms, désigne le temps nécessaire pour acheminer un volume élémentaire de données de la source jusqu'à la destination. Ce paramètre peut correspondre à une valeur maximale à ne pas dépasser, une mesure moyenne ou minimale ... mais en aucun cas ne désigne le temps total de transfert des données. Il est mesuré de bout en bout ou entre deux points de référence comme étant le temps nécessaire à l'acheminement d'une unité de volume, en général un paquet, entre les deux points de mesure. Il correspond au temps de transport proprement dit (ex. 5 μs/km pour la vitesse de propagation de la lumière dans une fibre optique) ajouté au temps de traversée des différents équipements.
- La gigue (jitter ou delay variation) qui désigne la variation du délai de transmission, exprimée en ms

Les ressources sont étroitement liées aux paramètres de QoS. En effet, une qualité de service donnée sera respectée si et seulement si les ressources influant sur le débit, la perte, le délai et la gigue sont bien disponibles. Si la ressource liée au débit est simple à définir, il n'en est pas de même pour les autres paramètres. Le délai et la gigue sont étroitement liés aux capacités d'acheminement du réseau en terme de longueur de connexion, de temps de transit dans les équipements ... Le taux de perte est lui lié aux capacités d'écoulement du trafic par le réseau.

En résumé, la QoS qui doit être assurée par l’allocation retenue entre plusieurs critères et pour cela voici un exemple de quelques contraintes qui permettent d’assouvir notre but de Qos:

- Le débit nécessaire à l'acheminement du flux de données des machines virtuelles instanciées sur un nœud physique doit être disponible sur le lien de sortie du routeur physique que ce soit au niveau de la fonction d'aiguillage ou dans les files d'attentes.

- Le SLA doit être respecté par l'application des paramètres fonctionnels et que les paramètres non fonctionnels correspondants à chaque réseau virtuel,

- Le temps de transit dans l'équipement et sur le lien de raccordement est le plus faible et le plus stable possible.

- Le flux de données est aiguillé dans la bonne Classe de Services pour être traité selon la priorité requise par l'application.

iii.Economie d'énergie :

Selon Gartner, le matériel informatique a consommé 830 térawatt heure (TWh) d'électricité en 2008 au niveau mondial et atteindra 900 TWh en 2012. En fonctionnant 24 heures sur 24, 7 jours sur 7, les nœuds physiques qui constituent la base des réseaux virtuels sont particulièrement gourmands en énergie, et sans ces nœuds plus de réseaux. Nous devons donc identifier leur consommation électrique et la minimiser sans pour autant diminuer les performances des réseaux virtuels qui y sont instanciés cela va de même pour la Qos.

d. Conclusion :

Nous avons par cette partie présenté les différentes contraintes que nous avons estimées importantes pour déterminer si une allocation est optimale et retourne des performances et une qualité de service préalablement entendue dans le SLA. Dans la partie suivante nous présentons notre implémentation via le logiciel le langage de programmation Java et le solver CHOCO.

Ceci dit la réalisation de l'expression de nos contraintes sur la plateforme de programmation java et avec l'utilisation de la librairie CHOCO a nécessité quelques changements non pas de fonds mais de formes ce qui nous a permis d'avoir une homogénéité et une ergonomie non négligeables dans la programmation des contraintes. A noter que pour un même problème, si plusieurs solutions sont possibles, Les

contraintes que nous voulons exprimer sont d'une grande complexité combinatoire, il faut donc réduire la complexité, pour cette raison la solution trouvée dépend de la façon dont le problème est déclaré.

4. Implémentation et tests :

Résoudre un CSP signifie qu'on peut chercher une solution ; toutes les solutions ou bien une solution qui optimise la fonction objective. En effet, pour explorer l'ensemble des configurations possibles et trouver celle qui convient aux besoins, on construit un "arbre de recherche". Ce dernier est construit en suivant une stratégie, dans la majorité des cas c'est la stratégie séparation et évaluation (branch and bound)[3]; le concept global de cette stratégie est d'arriver à éliminer des solutions partielles qui ne mènent pas à la solution désirée. De ce fait, on arrive souvent à obtenir la solution recherchée en des temps raisonnables; le risque d'éliminer toutes les solutions existe.

Pour résoudre un tel problème, il faudra procéder par essai. Si, à une certaine étape, on arrive à une impasse, il faudra revenir en arrière et remettre en doute un choix précédent. C'est pour ces raisons que notre solution utilise l'algorithme du Backtracking pour explorer l'arbre de recherche.

L'implémentation s'appuie sur les contraintes définies avec le CSP, le langage de programmation Choco nous permet d'exprimer le graphe de chaque réseau distinctement ainsi que les contraintes en matière de Qos qui le caractérisent, en prenant compte de chaque réseau et de ces caractéristiques fonctionnelles et non fonctionnelles exprimées dans le SLA ; notre programme nous donne la solution d'allocation la plus optimale.

La fonction d'allocation des liens virtuels sur les liens physiques est quant à elle assurée par les fonctionnalités d'Open Vswitch qui sont une implémentation logicielle d'un switch Ethernet. Concrètement il est constitué d'un service (ovs-vswitchd) et d'un module kernel (openvswitch_mod). Le service permet de commuter effectivement les

[3] Pour plus d'informations sur cette stratégie un cours de l'Université du Québec à Chicoutimi est mis en ligne à cette adresse: http://wwwens.uqac.ca/~rebaine/8INF806/techniquedebranchandboundcourshiver2005.pdf

paquets vers les bons ports virtuels, alors que le module kernel permet de capturer le trafic provenant des interfaces réseau, et d'y réinjecter le trafic légitime.

a. Implémentation du CSP :

Le processus de propagation de contraintes est l'outil algorithmique fondamental de la programmation par contraintes. Il systématise ce qui est déjà réalisé classiquement en recherche opérationnelle, en particulier en optimisation 0-1 avec les relations logiques, en propageant toute connaissance déduite (valeur d'une variable, réduction d'un domaine, réduction d'une contrainte) sur l'ensemble du système de contraintes. Ainsi, certaines contraintes mises en attente dites contraintes passives, peuvent devenir actives et par conséquent être exploitées à leur tour par les techniques de déduction de la connaissance (filtrage, réduction).

Pour implémenter notre CSP nous avons utilisé Choco. CHOCO est une librairie Java permettant de résoudre des problèmes combinatoires. C'est un langage de Programmation Par Contraintes, libre, en perpétuelle amélioration qui possède déjà de nombreuses fonctionnalités. CHOCO est initialement le noyau de la plate-forme OCRE: cette librairie Java définit les structures de bases d'un système de contraintes telles que les variables domaines, les contraintes, la propagation et la recherche arborescente.

Elle est construite sur un mécanisme de propagation basé sur des événements avec des structures backtrackable. Pour l'implémentation de notre projet, nous avons développé le module CSP qui doit en premier lieu exprimer les caractéristiques des réseaux virtuels en forme de CSP. Le CSP doit prendre en compte plusieurs points comme:

- Satisfaire les contraintes fonctionnelles en premier lieu. Si celles-ci ne sont pas satisfaites aucune allocation ne sera proposée.
- Garantir la QoS ainsi prévue par la SLA, en garantissant les paramètres non fonctionnels autant que cela soit possible.
- Faire économiser de l'énergie au fournisseur en allouant les réseaux virtuels sur des nœuds physiques avec des réseaux déjà instanciés et mettre en veille le reste des nœuds.

Notre solution est une solution centralisée, ainsi notre CSP a connaissance de toutes les ressources du réseau physique qu'elles soient allouées ou libres ; pour cela un cluster est mis en place avec toutes les informations nécessaires sur la capacité CPU et

mémoire de chaque nœud physique mais aussi des nœuds virtuels de chaque réseau virtuel qu'il soit déjà instancié (en exploitation) ou qu'il ne soit encore qu'une requête dans le système.

La problématique se posait aussi au niveau de la qualité de service, les questions que l'on devait se poser étaient :

- Comment connaitre les caractéristiques du réseau physique en temps réel ?
- Comment en connaitre les performances avant d'instancier un nouveau réseau virtuel ?
- Comment faire face à une coupure de lien ou à la saturation de ce dernier ?

En résumé, comment monitorer le trafic, collecter l'information mais la plus récente qui soit, mais aussi comment éviter une baisse des performances liées au délai de propagation.

Nous avons utilisé pour la collecte d'informations sur les performances du réseau l'outil I-Perf[4], il permet la mesure de différentes variables d'une connexion réseau IP. I-perf doit être lancé sur deux machines se trouvant de part et d'autre du réseau à tester. La première machine lance I-perf en "mode serveur" (avec l'option « -s »), la seconde en "mode client" (option « -c »). I-perf utilise par défaut le protocole TCP, nous préférons changer de protocole pour UDP, car les acquittements que renvoie la machine de destination dans le cadre du protocole TCP pourrait causer un Over Head[5] important dans le réseau ainsi pour minimiser cela nous utilisons UDP. Ainsi nous venons de constituer notre Cluster d'informations qui nous fournira les valeurs et mesures des paramètres Qos dont nous avons besoin.

On doit aussi assurer des conditions implicites qui nous font exprimer en langage Choco, ces conditions sont :

- Routeur virtuel alloué une et une seule fois :
- Capacité CPU routeur physique doit être supérieure ou égale à la somme des capacités CPU des routeurs virtuels qui lui sont alloués :

[4] Pour plus d'informations consulter http://www.iperf.fr/
[5] Surcharge supplémentaire dans le trafic réseau.

- Capacité mémoire du routeur physique supérieure ou égale à la somme des capacités CPU des routeurs virtuels qui lui sont alloués :

Pour rappel le tableau 6 (Mapping SLA et Service), mettait en lumière les différents types de Qos, dépendamment du service fourni par son réseau virtuel, Il fallait trouver une méthode claire et la plus simple possible pour exprimer cette différence et l'importance de l'ordre des principaux paramètres non fonctionnels que doit prendre en compte la solution pour proposer une allocation la plus optimale possible.

Pour cela on a mis en place un système de poids. En effet, chaque paramètre possède un poids, son poids diffère par rapport au service et au SLA qui lui correspondent. Nous reprenons pour cela le Tableau 7 (Classement des critères par service) dont les valeurs seront celles sur lesquelles on va se baser pour proposer des valeurs de poids. Nous obtenons le tableau suivant :

	Taux de pertes	Délai	Gigue	Disponibilité
VOIP	2	4	3	1
Vidéophone	2	4	3	1
Téléphonie	2	3	1	4
Multimédia	3	4	1	2
VOD	2	4	1	3
VPN	2	4	1	3
Données temps réels	3	4	1	2
Données	4	3	1	2
Streaming	4	2	1	3

Tableau 8 : Poids des paramètres par Service

NB : Le système de poids a été mis en place pour les paramètres fonctionnels qui ne sont pas négociables, pour cette raison nous avons attribué la plus grande valeur de poids aux trois paramètres fonctionnels et qui est la même peu importe le service proposé par le réseau virtuel :

La valeur poids (CPU, Mémoire, Débit) = 10.

b. Aspect Green (économie d'énergie) :

En théorie nous avons mis en place une règle pour le calcul de l'utilité Provider, mais celle-ci ne peut être appliquée et testée dans un contexte de recherche ; il est à rappeler que cette utilité Provider prenait en compte la consommation énergétique des équipements qui étaient alloués à chaque réseau virtuel et ce faisant, calculer la consommation globale d'un réseau virtuel et tenter de la minimiser pour optimiser le gain. Dans un même but et dans un souci d'améliorer la solution proposée en gardant comme objectif le fait d'exprimer cette condition en mode CSP ainsi que l'économie d'énergie, nous avons opté pour une solution différente. On a intégré une fonction $Score\ global$ qui est égale au score de chaque paramètre multiplié par son poids. Ce qui donne :

$$Score\ global = Score\ fonctionnel + Score\ Non\ Fonctionnel$$

Nous avons eu l'occasion précédemment d'exposer la méthode de calcul des poids de chaque paramètre, il nous reste donc à exposer celle du scoring mis en place.

i.Scoring Fonctionnel :

Le score des paramètres fonctionnels est une valeur globale qui est égale au nombre de routeurs physiques alloués peu importe le service ou le paramètre. De là nous pouvons conclure que :

$$Score\ fontionnel = \sum_{1}^{j} hUse[j] * 10$$

ii.Scoring Non Fonctionnel :

Le calcul du score Non Fonctionnel est égal à la somme des scores individuels de chaque paramètre non fonctionnel que l'on a appelé *Impact* multiplié par la valeur de son poids.

$$Score\ non\ fontionnel = \sum_{1}^{4} Impact_i \times Poids_i$$

Nous avons mis en place une valeur nommée *Impact* qui calcule l'impact d'une allocation donnée sur chaque paramètre non fonctionnel. Nous utilisons pour cela la matrice *P* et la matrice *D,* contenant respectivement le potentiel pour un couple de routeurs physiques (h_j,h_{j2}) et la consommation de potentiel pour un couple de routeurs

virtuels (V_i,V_{i2}). De la on s'intéresse aux routeurs physiques actifs. Le calcul de la valeur *Impact* se fait comme suit :

- X =le montant du couple de serveurs virtuels = D[i][i2]
- Y =le montant du couple de serveurs physiques= P[j][j2]
- $Z = \begin{cases} 0 & si\ X \leq Y \\ X - Y & sinon \end{cases}$
- $Impact_i = Max\ (Z)$

NB : Par conséquent, il est à rappeler qu'il existe un Impact pour chaque paramètre non fonctionnel.

Pour conclure, le score global est l'agrégation pondérée des scores (fonctionnels et non fonctionnels). Ainsi, pour limiter l'usage abusif des routeurs physiques, une fonction objective a été mise en place dans le but de minimiser la variable Score global.

5. Conclusion :

Dans ce projet nous avons réalisé une solution à base de CSP qui a pour fonction l'allocation dynamique de ressources dans les réseaux virtuels avec possibilité de reconfiguration. Nous avons identifié ce problème comme étant un problème de matching de graphe. L'une des méthodes de résolution des problèmes combinatoires est de les exprimer en CSP. Nous avons utilisé cet outil mathématique pour formuler le problème d'allocation de ressources ainsi que les besoins en QoS et en économie d'énergie

La séparation du problème d'allocation de ressources en deux parties, l'une incluant les contraintes fortes représentées par les paramètres fonctionnels et l'autre l'expression de la Qualité de Service avec des contraintes moins fortes mais respectant le SLA, nous a permis d'avoir des résultats encourageants et qui sont optimaux dûs à leur résolution par un solveur CSP (Choco).

L'exposition combinatoire de solution et d'ensemble de recherche nous laisse croire que notre solution ne passerait pas à l'échelle, mais il faut considérer que d'une part, certaines zones de recherche sont inaccessibles car les contraintes n'y sont pas respectées (en effet dès qu'une étape viole une contrainte, l'algorithme opère un backtrack sans descendre dans le sous arbre), et d'autre part des stratégies heuristiques d'exploration peuvent être utilisées afin d'obtenir des résultats le plutôt possible dans la recherche.

Bibliographie:

[1] "Five Best Practices to Protect Your Virtual Environment," Juniper Network, 2012.
[2] "Security as a Service (SecaaS) Defined categories of service," 2011.
[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 42–57, 2013.
[4] S. Yu, Y. Tian, S. Guo, and D. Wu, "Can We Beat DDoS Attacks in Clouds?," in *Parallel and Distributed Systems, IEEE Transactions on*, 2013, vol. PP, pp. 1–1.
[5] W. Yassin, N. I. Udzir, Z. Muda, A. Abdullah, and M. T. Abdullah, "A cloud-based intrusion detection service framework," in *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*, 2012, pp. 213–218.
[6] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2011, pp. 31–32.
[7] Y. Meng, W. Li, and L.-F. Kwok, "Design of Cloud-Based Parallel Exclusive Signature Matching Model in Intrusion Detection," in *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*, 2013, pp. 175–182.
[8] Y. Meng, W. Li, and L.-F. Kwok, "Towards adaptive false alarm reduction using Cloud as a Service," in *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, 2013, pp. 420–425.
[9] M. Darwish, A. Ouda, and L. F. Capretz, "Cloud-based DDoS attacks and defenses," in *2013 International Conference on Information Society (i-Society)*, 2013, pp. 67–71.
[10] T. Jirsik, M. Husak, P. Celeda, and Z. Eichler, "Cloud-based security research testbed: A DDoS use case," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–2.
[11] V. Getov, "Security as a Service in Smart Clouds – Opportunities and Concerns," in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*, 2012, pp. 373–379.
[12] D. Krishnan and M. Chatterjee, "Cloud security management suite x2014; Security As a Service," in *Information and Communication Technologies (WICT), 2012 World Congress on*, 2012, pp. 431–436.
[13] E. G. Amoroso, Cyber attacks: protecting national infrastructure. Elsevier, 2012.
[14] A. Abdel-Aziz and J. Esler, "Intrusion Detection & Response - Leveraging Next Generation Firewall Technology," SANS - Institue, 2009.
[15] R. Lua and K. C. Yow, "Mitigating ddos attacks with transparent and intelligent fast-flux swarm network," *Netw. IEEE*, vol. 25, no. 4, pp. 28–33, 2011.
[16] M. H. Sqalli, F. Al-Haidari, and K. Salah, "Edos-shield-a two-steps mitigation technique against edos attacks in cloud computing," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 2011, pp. 49–56.
[17] P. Du and A. Nakao, "DDoS defense as a network service," in *Network Operations and Management Symposium (NOMS), 2010 IEEE*, 2010, pp. 894–897.

[18] J. Chen, Y. Wang, and X. Wang, “On-Demand Security Architecture for Cloud Computing.,” *IEEE Comput.*, vol. 45, no. 7, pp. 73–78, 2012.
[19] A. R. Khakpour and A. X. Liu, “First Step toward Cloud-Based Firewalling,” in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*, 2012, pp. 41–50.
[20] S. Subashini and V. Kavitha, “Review: A survey on security issues in service delivery models of cloud computing,” *J Netw Comput Appl*, vol. 34, no. 1, pp. 1–11, Jan. 2011.
[21] A. Bouhoula, Z. Trabelsi, E. Barka, and M.-A. Benelbahri, “Firewall filtering rules analysis for anomalies detection,” *Int. J. Secur. Netw.*, vol. 3, no. 3, pp. 161–172, 2008.
[22] J. E. Canavan, *The Fundamentals of Network Security*. Artech House, 2001.
[23] S. Suehring and R. Ziegler, *Linux Firewalls (Novell Press)*. Novell Press, 2005.
[24] J. Garcia-Alfaro, F. Cuppens, N. Cuppens-Boulahia, S. Martinez, and J. Cabot, “Management of stateful firewall misconfiguration,” *Comput. Secur.*, no. 0, 2013.
[25] J. Pescatore and G. Young, “Defining the Next-Generation Firewall,” *Gart. RAS Core Res. Note Httpwww Ga1tner Com*, 2009.
[26] “SecaaS Implementation Guidance Category 10 Network Security,” Cloud Security Alliance, 2012.
[27] A. X. Liu, *Firewall design and analysis*, vol. 4. World Scientific, 2011.
[28] R. N. Smith, Y. Chen, and S. Bhattacharya, “Cascade of distributed and cooperating firewalls in a secure data network,” *Knowl. Data Eng. IEEE Trans. On*, vol. 15, no. 5, pp. 1307–1315, 2003.
[29] E. W. Fulp and R. J. Farley, “A Function-Parallel Architecture for High-Speed Firewalls,” in *IEEE ICC*, 2006, vol. 5, pp. 2213–2218.
[30] L. Xiaofei, “Analysis and design of distributed firewall system in campus network,” in *Computational Intelligence and Industrial Applications, 2009. PACIIA 2009. Asia-Pacific Conference on*, 2009, vol. 2, pp. 468–471.
[31] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, “A view of cloud computing,” *Commun. ACM*, vol. 53, p. 50, Apr. 2010.
[32] J. Dittrich and M. Braun, *Business Process Outsourcing*. Schäffer-Poeschel, 2004.
[33] D. Basak, R. Toshniwal, S. Maskalik, and A. Sequeira, “Virtualizing networking and security in the cloud,” *SIGOPS Oper Syst Rev*, vol. 44, no. 4, pp. 86–94, Dec. 2010.
[34] E. L. Haletkyl, Are Virtual Firewalls a Real Solution for VM Security? 2013.
[35] G. E. Gonçalves, P. T. Endo, T. Damasceno, A. V. de A. P. Cordeiro, D. Sadok, J. Kelner, B. Melander, and J.-E. M\a angs, “Resource allocation in clouds: concepts, tools and research challenges,” *XXIX SBRC-Gramado-RS*, 2011.
[36] L. Mashayekhy and D. Grosu, “A coalitional game-based mechanism for forming cloud federations,” in *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, 2012, pp. 223–227.
[37] X. Nan, Y. He, and L. Guan, “Optimal resource allocation for multimedia cloud based on queuing model,” in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, 2011, pp. 1–6.
[38] N. R. R. Mohan and E. B. Raj, “Resource Allocation Techniques in Cloud Computing–Research Challenges for Applications,” in *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*, 2012, pp. 556–560.
[39] V. C. Emeakaroha, I. Brandic, M. Maurer, and I. Breskovic, “SLA-Aware application deployment and resource allocation in clouds,” in *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, 2011, pp. 298–303.

[40] P. S. Pillai and S. Rao, “Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory.”
[41] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, “A game-theoretic method of fair resource allocation for cloud computing services,” *J. Supercomput.*, vol. 54, no. 2, pp. 252–269, 2010.
[42] K. G. Srinivasa, S. Srinidhi, K. S. Kumar, V. Shenvi, U. S. Kaushik, and K. Mishra, “Game theoretic resource allocation in cloud computing,” in *Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*, 2014, pp. 36–42.
[43] W. Wu, X. Zhang, Y. Zheng, and H. Liang, “Agent-based layered cloud resource management model,” in *Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on*, 2013, vol. 2, pp. 70–74.
[44] D. Talia, “Clouds meet agents: Toward intelligent cloud services,” *IEEE Internet Comput.*, vol. 16, no. 2, pp. 78–81, 2012.
[45] M. Habiba, M. Islam, A. B. M. Ali, and others, “Access Control Management for Cloud,” in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 2013, pp. 485–492.
[46] A. M. Talib, R. Atan, R. Abdullah, and A. Murad, “Security framework of cloud data storage based on Multi Agent system architecture-A pilot study,” in *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on*, 2012, pp. 54–59.
[47] M. Habiba and S. Akhter, “MAS workflow model and scheduling algorithm for disaster management system,” in *Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on*, 2012, pp. 164–173.
[48] N. Khezami, S. Otmane, M. Mallem, and others, “A new formal model of collaboration by multi-agent systems,” in *International Conference Integration of Knowledge Intensive Multi-Agent Systems (KIMAS 2005)*, 2005, pp. 32–37.
[49] P. Urien, E. Marie, and C. Kiennert, “A New Convergent Identity System Based on EAP-TLS Smart Cards,” in *Network and Information Systems Security (SAR-SSI), 2011 Conference on*, 2011, pp. 1–6.
[50] P. Urien, E. Marie, and C. Kiennert, “An innovative solution for cloud computing authentication: Grids of eap-tls smart cards,” in *Digital Telecommunications (ICDT), 2010 Fifth International Conference on*, 2010, pp. 22–27.
[51] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, and others, “Extensible authentication protocol (EAP),” RFC 3748, June, 2004.
[52] *Netfilter*. 2013.
[53] *Hping-3*. 2013.
[54] *Iperf*. 2013.
[55] A. Khiyaita, M. Zbakh, H. El Bakkali, and D. El Kettani, “Load balancing cloud computing: State of art,” in *Network Security and Systems (JNS2), 2012 National Days of*, 2012, pp. 106–109.
[56] G. et al, “The title Here,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng. 312*, p. 480, Dec. 2013.
[57] F. Bellifemine, A. Poggi, and G. Rimassa, “Developing multi-agent systems with JADE,” in *Intelligent Agents VII Agent Theories Architectures and Languages*, Springer, 2001, pp. 89–103.
[58] M.-P. Huget, “FIPA website,” Mar-2014. [Online]. Available: http://www.fipa.org/.

[59] T. Kunz, “The influence of different workload descriptions on a heuristic load balancing scheme,” *Softw. Eng. IEEE Trans. On*, vol. 17, no. 7, pp. 725–730, Jul. 1991.
[60] A. Haider, R. Potter, and A. Nakao, “Challenges in resource allocation in network virtualization,” in *20th ITC Specialist Seminar*, 2009, vol. 20, p. 22.
[61] Y. Zhu and M. Ammar, “Algorithms for assigning substrate network resources to virtual network components,” in *Proc. IEEE INFOCOM*, 2006.
[62] J. Lu and J. Turner, “Efficient mapping of virtual networks onto a shared substrate,” *Wash. Univ. USA Tech Rep*, 2006.
[63] I. Houidi, W. Louati, and D. Zeghlache, “A Distributed and Autonomic Virtual Network Mapping Framework,” in *Autonomic and Autonomous Systems, International Conference on*, Los Alamitos, CA, USA, 2008, pp. 241–247.
[64] N. F. Butt, M. Chowdhury, and R. Boutaba, Topology-awareness and reoptimization mechanism for virtual network embedding. Springer, 2010.
[65] N. M. M. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *Commun. Mag. IEEE*, vol. 47, no. 7, pp. 20–26, 2009.
[66] C. Desrosiers, “Le matching de graphes: une introduction.” Ecole Polytechnique de Montréal, Mar-2006.
[67] Long-Tae Park, Jong-Wook Baek, and J. Woon-Ki Hong, “Management of service level agreements for multimedia Internet service using a utility model,” *IEEE Commun. Mag.*, vol. 39, no. 5, pp. 100–106, May 2001.
[68] E. Marilly, O. Martinot, H. Papini, and D. Goderis, “Service level agreements: a main challenge for next generation networks,” in *2nd European Conference on Universal Multiservice Networks, 2002. ECUMN 2002*, 2002, pp. 297– 304.
[69] F. Aleskerov, D. Bouyssou, B. Monjardet, D. Bouyssou, and B. Monjardet, *Utility maximization, choice and preference*, vol. 16. Springer, 2007.
[70] C. Solnon, “Résolution de problèmes combinatoires et optimisation par colonies de fourmis.” Université Lyon 1.
[71] J. Teghem and M. Pirlot, Optimisation approchée en recherche opérationnelle recherches locales, réseaux neuronaux et satisfaction de contraintes. Paris: Hermès Science publications, 2002.
[72] E. Tsang, *Foundations of Constraint Satisfaction*, Computation in Cognitive Science. Departement of Computer Science, University of Essex, UK: Academic Press, 1993.